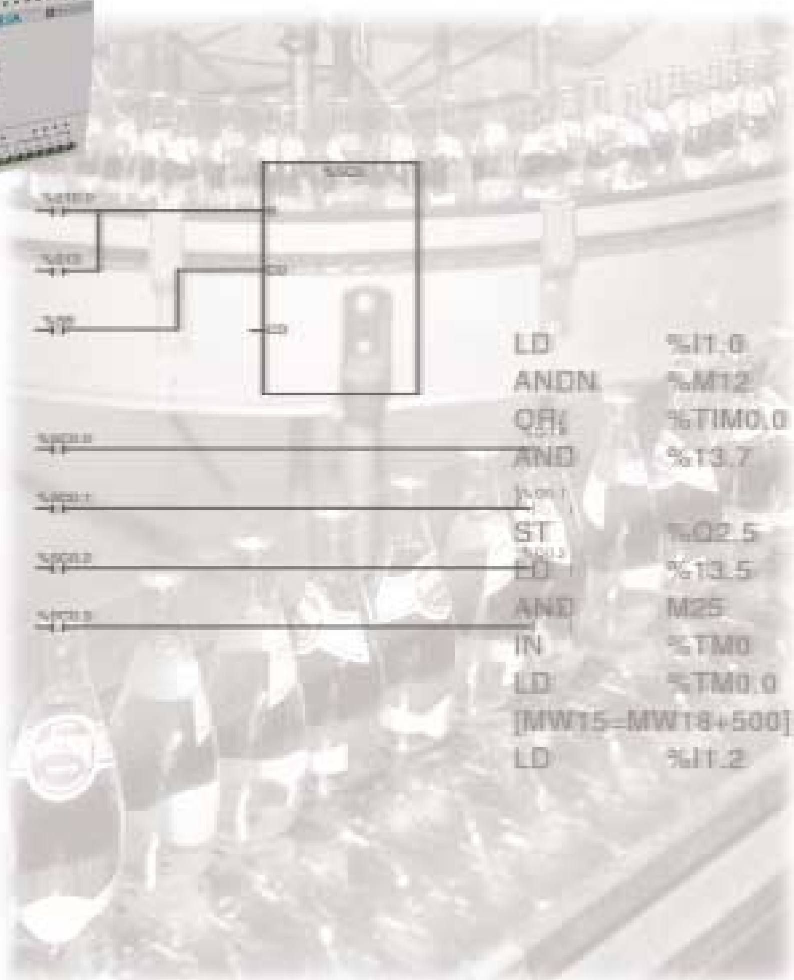


We Make Automation Easier for You

Modicon TSX Neza 可编程控制器
指令集和通讯





Merlin Gerin
Modicon
Square D
Telemecanique

Schneider Electric
Building a New Electric World

序言

Modicon TSX Neza 可编程控制器操作手册系列包括 A、B、C 三部分：

A 部分	硬件介绍	《产品指南》
B 部分	指令集和通讯	《指令集和通讯》（本手册）
C 部分	编程软件	《PL707 WIN 编程软件操作手册》

B部分包括两个层次的信息：

- 第一层次的信息帮助客户实现简单的功能。在该情况下用户只需阅读文章中的灰体字部分。
- 第二层次的信息帮助用户实现 TSX Neza 的所有功能。在该情况下用户需阅读所有内容。

目录

B

Page

1.概述

1.1 总论

PLC 所执行的控制命令由用户编写的控制程序来决定。编写 TSX Neza PLC 的控制程序需要使用 Neza PLC 所支持的编程软件 PL707WIN。

PL707WIN 编程软件支持指令列表和梯型图编程语言。指令列表语言是一种基于行级的文件型类布尔语言，它也可以用于处理数字操作。梯形图语言是一种基于梯级的图形布尔语言。PL707WIN 还允许在指令列表语言和梯形图语言之间进行转换。

TSX08PALM HT01/05 掌上电脑编程器 (PPC) 可用于对 Neza PLC 编程。

1.2 指令列表(List 或 IL)

程序结构

一个用 PL707WIN 语言所写的程序包括一系列的不同类型的指令(1000 多条指令)
每个语句行都有一个自动生成的编号，一个指令代码和一个位类型或字类型的操作数。

指令举例

003 LD %I0.1

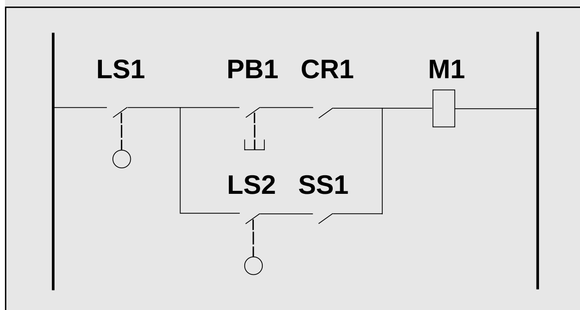
编号 — 003；指令代码 — LD；操作数 — %I0.1

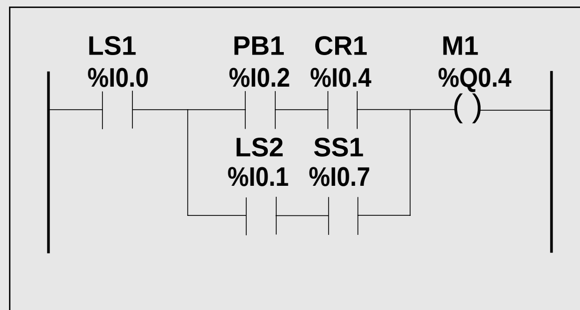
梯形图是逻辑表达式的一种图形化表现方式。它可由指令列表程序替代。指令列表程序是指一系列由布尔指令序列构成的逻辑表达式。

指令		
指令类型	举例	
• 位指令	004 LD%M10	读内部位 %M10
• 块指令	008 LD%TM0	打开定时器 %TM0
• 字指令	010 [%MW10:=%MW50+100]	加运算
• 程序指令	015 SR5	调用子程序 #5

1.3 梯形图

梯形图类似于用来描述继电器控制电路的逻辑图。主要的不同是, 在梯形图编程中所有的输入都由触点符号（ -| |- ）表示, 所有的输出都由线圈符号（ -( ) -)表示, 并且在梯形图指令集中包括数字运算。



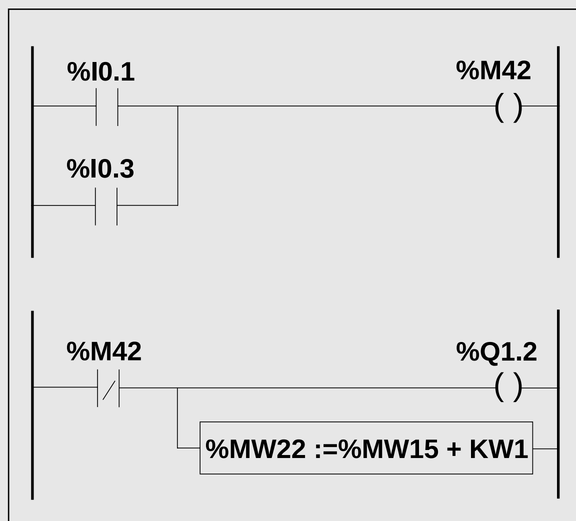


上图例是一个继电器逻辑电路的简化电路图和它的等效梯形图。请注意, 梯形图中的每个输入与继电器逻辑图中的开关设备相关就以触点形式表示。继电器逻辑图中的 M1 输出线圈在梯形图中用输出线圈符号表示。梯形图中每个触点／线圈符号上的地址标号对应于PLC 相连的外部输入／输出的位置。

用梯形图编写的程序由梯级构成—梯级是指画在象征电势的两条垂直栏里的特定图形指令集—并由 PLC 按顺序执行。图形指令集用于表示：

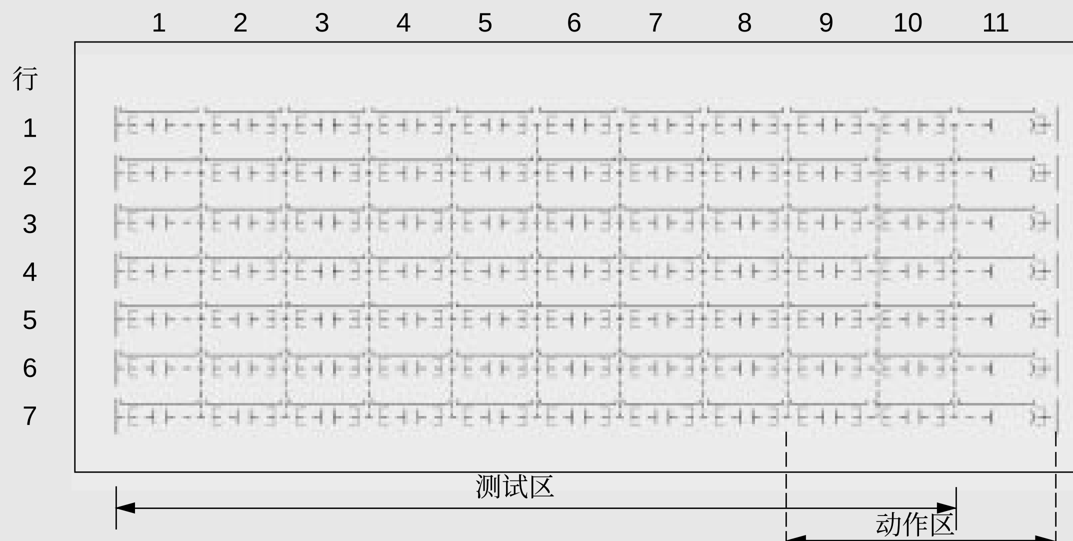
- PLC 的输入／输出（按钮、传感器、继电器、指示灯……）
- PLC 的功能（定时器、计数器……）
- 算术和逻辑运算（加法、除法、与、或……）
- 比较运算和其他数字运算（A<B、A=B、移位、循环……）
- PLC 的内部变量（位、字……）

竖直和水平连接这些图形指令从而实现一个或多个输出的与／或动作。
一个梯级只能支持一组相关指令。
例如，这个梯形图由两个梯级组成



1.3-1 编程原则

每个梯形图的梯级由 7 行 11 列组成，划分为两个区域：



- 测试区，它包括动作发生所必须具备的条件。
- 动作区，它包括由相关测试引起的输出或操作。

梯级形象化为 7 行 11 列的编程网格，并从最左上方的一个网格开始。在测试区编写测试指令、比较程序和功能程序, 其中测试指令应该左对齐。测试逻辑将使得动作区中的线圈程序、数字运算和程序流控制指令保持连贯性。动作指令应该右对齐。梯级自上而下自左而右地执行（进行测试和计算输出）。

除了梯级以外，在它上方还有一个梯级注释区。
梯级注释区用于说明梯级的逻辑目的。它包括梯级编号，所有标号 （%Li:） 或子程序说明（SRi:），梯级标题和梯级注解。关于梯级注释区和相应指令列表行注释的详细说明，见节B.1.3-3。

- 触点、线圈和程序流指令

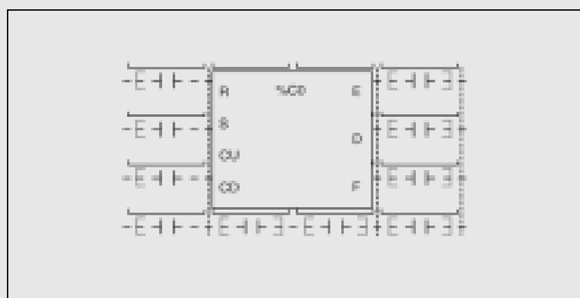
触点，线圈和程序流（跳转和子程序调用）指令占用编程网格中的一格。功能块、比较块和操作块占用多格。



- 功能块

功能块位于编程网格的测试区。它必须写在第一行；在它的上面不可以有梯形图指令或连接线。梯形图测试指令导入功能块的输入边，并且测试指令中的与／或动作指令由功能块的输出边导出。

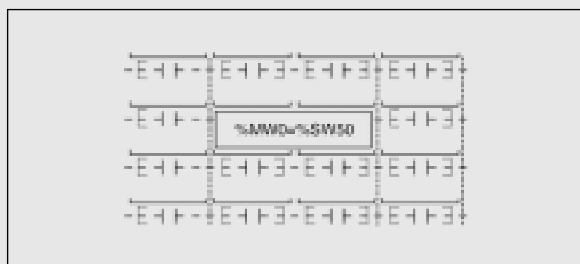
功能块是垂直指向的，它占有四行两列编程网格。



- 比较块

比较块位于编程网格中的测试区。它可以写在测试区的任意行列，只要指令全长不超出测试区。

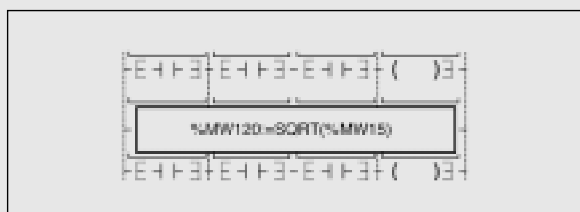
比较块是水平走向的。它占有一行两列编程网格。



- 操作块

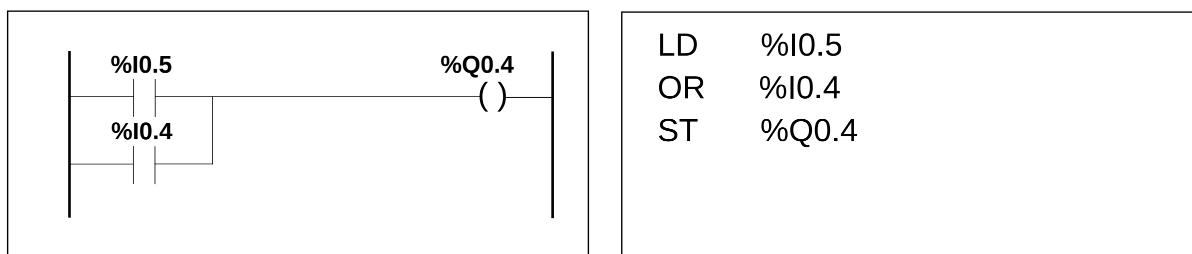
操作块位于编程网格的动作区。它可以写在动作区的任意一行。指令应该右对齐，所以它写在右边直至最后一列。

操作块是水平指向的。它占有一行四列编程网格。



1.3-2 梯形图与指令列表的可逆性

在本手册中, 可逆性是指用于 TSX Neza 的编程软件 PL7 07 WIN 可将 TSX Neza 的应用程序在梯形图编程语言和指令列表语言之间相互转换。可以通过设定来选择任意一种语言编写 PL707WIN 程序。也可以通过工具菜单中的梯形图／指令列表切换选项来完成个别梯形图梯级与指令列表之间的相互转换。



要了解可逆性就应该清楚“ 梯级 ”和“ 序列 ”之间的关系。“ 梯级 ”是由梯形图编程指令集合组成的逻辑表达式, “ 序列 ”则是由指令列表编程指令集合组成的。上图左边举例说明了一个在用户梯形图程序中使用的通用梯级。右边是由指令列表序列表示的等效程序逻辑图。

本质上来说，不论用梯形图还是指令列表语言编写程序，内部都是以指令列表形式存储的。PL707WIN 的一个优点是两种语言的程序结构相似。另外，虽然内部存储的指令是程序的指令列表形式, 但可根据用户的选择显示指令列表／梯形图两种不同视图, 也可按两种不同的方式进行编辑。其中指令列表是程序的基本形式, 所以所有的梯形图都可以转换成指令列表, 但是有些指令列表逻辑不能转换成梯形图。为了保证从指令列表向梯形图的可逆性，必须参照节 B.1.3-3 中的指令列表编程约定。

1.3-3 可逆的指令列表编程约定

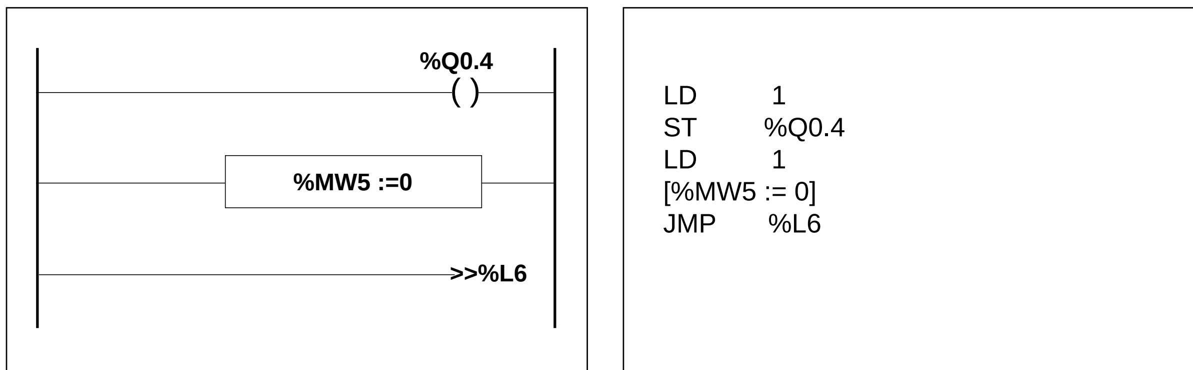
用指令列表语言编写的可逆功能块结构需要使用一些特定的指令。它们是：

- BLK（标志功能块的开始，定义梯级和功能块输入部分的起始）
- OUT_BLK（标志功能块输出部分的起始）
- END_BLK（标志功能块和梯级的结束）

指令列表程序不是一定要使用可逆的功能块才能正常运行。在编写指令列表程序时可使用一些不可逆的指令。在节 B2.2 的功能块说明部分将介绍不可逆指令列表编程。另一个重要的约定是避免使用某些指令列表指令，并避免某些指令和操作数的结合，因为在梯形图中没有相应的等效值。例如，N 指令（位 Not 取布尔累加器的反值）就没有等效的梯形图指令。下面的表格列出了所有不能转换成梯形图的指令列表指令。

指令列表指令	操作数	说明
JMPCN	%Li	条件为非时跳转
N	没有	取反 (NOT)
ENDCN	没有	条件为非时结束
XORN	任何数	XOR 之后进行 NOT 运算

无条件梯级也同样遵守指令列表编程约定以保证指令列表和梯形图之间的可逆性。无条件梯级是指没有测试或条件的梯级，输出指令在任何情况下都能被激活或执行。下图列出了无条件梯级和对应的等效指令列表序列。



请注意，除了一条特殊指令以外，所有的无条件指令列表序列都以紧跟着数字 1 的装入指令开始。这意味着将布尔累加器置 1 ，从而每次程序扫描时线圈（存储指令）都置 1 ，%MW5 都置 0 。上面提及的特殊指令是指无条件跳转指令。这条指令列表指令的执行与布尔累加器值无关，所以就不用像上面两个例子那样将累加器置 1 。

如果要将一个非完全可逆的指令列表程序转换成梯形图，则其可逆部分将以梯形图的形式显示，而不可逆部分将以“ 梯形图指令列表梯级 ”显示。梯形图指令列表梯级就像一个小的指令列表编辑器，使你可以看到并修改梯形图程序中的不可逆部分。

程序注释

指令列表编辑器允许使用指令列表注释行来注释程序。这些注释可与编程指令同行, 也可以另外起行。梯形图编辑器允许直接使用梯级上方的梯级注释区进行注释。

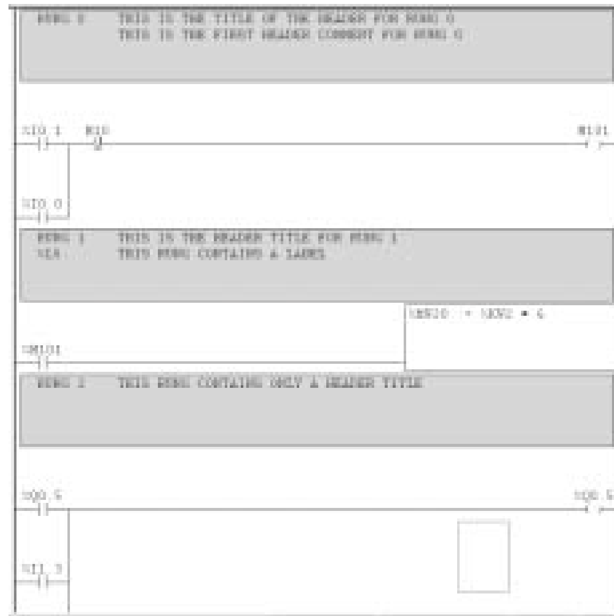
PL707WIN 的可逆性也表现在注释上。当把一个指令列表程序转换成梯形图时，PL707WIN 会在梯级注释区内写入指令列表注释。简而言之，指令列表序列中的注释对应梯级注释区。

指令列表序列结束后的第一个注释行是用来做梯级标题的（该行无指令）。其后的注释（如果存在的话），是可用于梯级注释区的主体注释。如果存在主体注释，则可忽略指令列表序列间的注释行，因为这类注释行通常还包含了指令列表指令。



注意：注释可用中文输入，但两侧的括号和 * 号 “（*. . .*）” 需为英文输入状态（半角状态）。

当含有梯级注释区的梯级转换成指令列表, 注释区中的注释将被插入指令列表序列之间。所有标号和子程序说明（%Li：或 SRi：）紧跟在注释区之后, 并在指令列表序列开始之前。如果梯级是由指令列表转换而来，则会忽略注释，但在指令列表编辑器中会恢复在原转换中被忽略的注释。



2.指令集

2.1 布尔指令

2.1-1 主要位对象定义

- **I/O** 位

该类位的地址系统是 I/O 电状态的“逻辑映像”，被存在数据存储器中，每次程序扫描时进行更新。

- 内部位

内部位是用户使用程序时内部 I/O 存储区域。

注意：未处于使用状态的 I/O 位不可以用作内部位

- 系统位

系统位 %S0 到 %S127 用于监控 PLC 及应用程序的正常运行。这类位的用途和使用方法见 B 部分第 5 节。

- 字抽取位：见节 **3.1-1**

操作数位列表

下表列出了所有类型的操作数位。

类型	地址（或值）	最大个数	写访问**(1)**	见节
立即值	0 或 1	_	_	_
输入位 输出位	%Ii.j. (2) %Qi.j (2)	48 32	不是 是	A 部分 A 部分
内部位	%Mi	128 (3)	是	
系统位	%Si	128	根据 i	5.1
功能块位	%TMi.Q, %DRi.F 等	_	不是(4)	2.2-1
可逆功能块位	E,D,F,Q,TH0, TH1		不是	3.3-1
字抽取位	可变	可变	可变	3.1-1

(1) 在程序中写或者由终端在数据编辑器中写。

(2) 当 i = 0 为本体 PLC 或对等 PLC，i=1-3 为 1-3 号扩展 I/O，j=I/O 点。I/O 位可在数据编辑器中置为 0 或 1。

(3) 如果发生电源断电则保存前 64 位。

(4) 除了 %SBRi.j 和 %SCi.j，这些位都是可读写的。

2.1-2 布尔指令说明

布尔指令可与梯形图语言元素相比较。

测试元件，例如：LD（装入）指令等效于常开触点。

LD %I0.0　　　　若控制位为状态 1 ，则闭合常开触点。

动作元件，例如：ST（存储）指令等效于一个线圈。

ST %Q0.0　　　　相关位对象取相应累加器的位逻辑值（前一次逻辑运算的结果）

布尔等式：

LD %I0.0　　　　将测试元件的布尔值用于动作元件。

AND %I0.1

ST %Q0.0

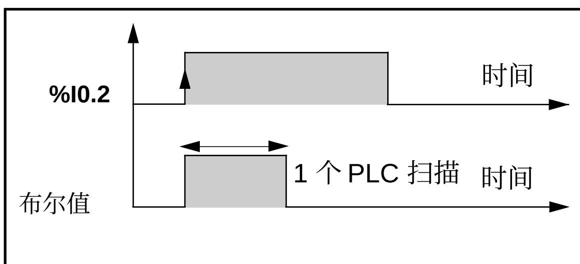
上升沿和下降沿

测试指令可用于检测 PLC 输入的上升或下降沿。当第 n 次扫描所得输入与第 n-1次不同，且在该扫描中保持不变，就表明检测到一个沿。

LDR 指令（R：上升沿）等效于上升沿触点：

LDR %I0.0　　%I0.0 ─┤P├─ (1)

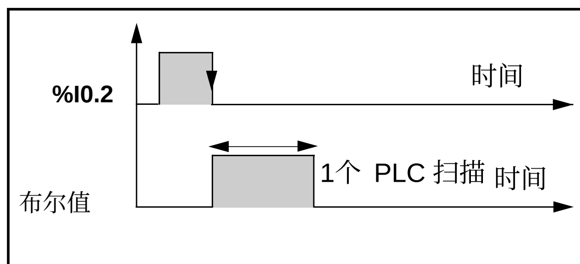
上升沿：检测到当前控制输入从 0 变到 1。



LDF 指令（F：下降沿）等效于一个下降沿触点：

LDR %I0.0　　%I0.0 ─┤N├─(2)

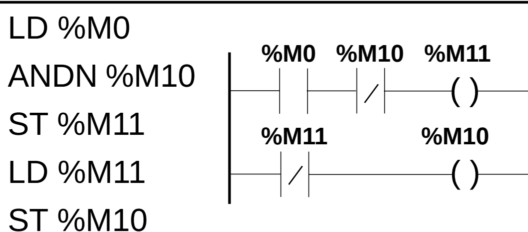
下降沿：检测到当前控制输入从 1 变到 0。



上升沿和下降沿指令不仅可以应用于输入 %I，还可应用于内部位（前 32 个，%M0～%M31）来检测其它位（或布尔值）的边沿。

右边是一个例子，位 %M11 检测到位 %M0 的上升沿。

LD %M0
ANDN %M10
ST %M11
LD %M11
ST %M10



(1) 上升沿跳变触点

(2) 下降沿跳变触点

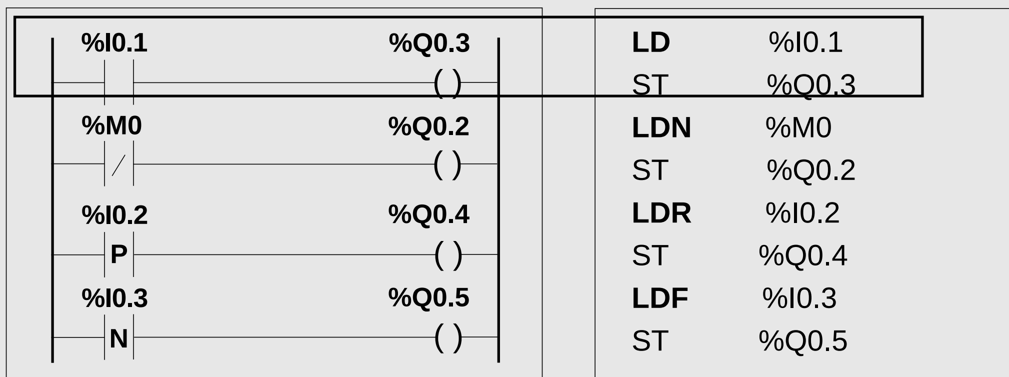
(3) 当发生冷启动或热启动时，即使输入保持 1，应用程序也会测到一个上升沿。为消除该现象，要用指令 LD %S1 和 ENDC 作为程序的开始。

指令格式

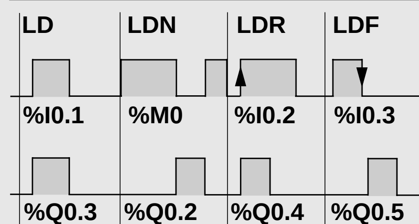
布尔指令用粗体字表示。每个等式由相应的梯形图表示。

装入指令 **LD LDN LDR LDF**

指令 LD、LDN、LDR 和 LDF 分别对应于常开、常闭、上升沿和下降沿触点。



指令	操作数
LD	0/1,%I,%Q,%M,%S,%BLK.x,%*:Xk,[
LDN	%I,%Q,%M,%S,%BLK.x,%*:Xk,[
LDR	%I %M
LDF	%I %M



操作数列表

0/1	立即值 0 或 1
%I	PLC 输入 %Ii.j
%Q	PLC 输出 %Qi.j
%M	内部位 %Mi
%S	系统位 %Si
%BLK.x	功能块位，列如 %TMi.Q
%*:Xk	字抽取位, 列如%MWi:Xk
[	比较表达式 例如[%MWi<1000]

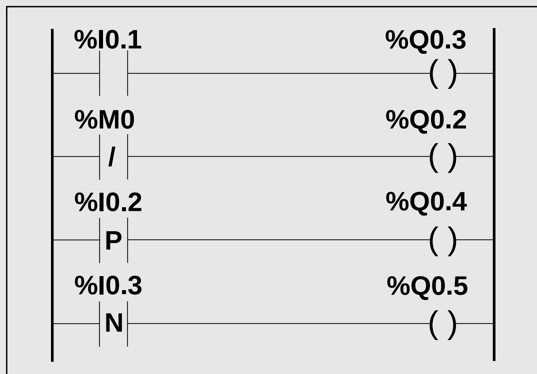
时序图

四张时序图组合在一起。



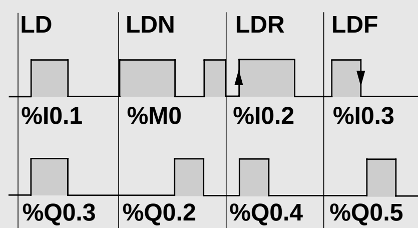
2.1-3 装入指令 LD LDN LDR LDF

指令 LD、LDN、LDR 和 LDF 分别对应常开、常闭、上升沿和下降沿触点。(LDR 和 LDF 只用于 PLC 的输入和前 32 个内部位，%M0～%M31)。



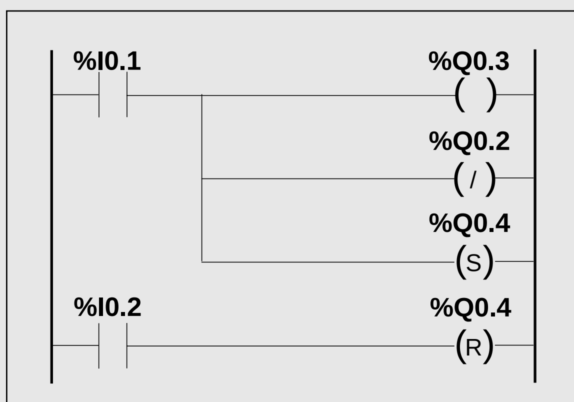

```
LD     %I0.1
ST     %Q0.3
LDN    %M0
ST     %Q0.2
LDR    %I0.2
ST     %Q0.4
LDF    %I0.3
ST     %Q0.5
```

指令	操作数
LD	0/1,%I,%Q,%M,%S,%BLK.x,%*:Xk,[
LDN	%I,%Q,%M,%S,%BLK.x,%*:Xk,[
LDR	%I %M
LDF	%I %M



2.1-4 赋值指令 ST STN S R

指令 ST、STN、S 和 R 分别对应直接、取反、置位和复位的线圈。



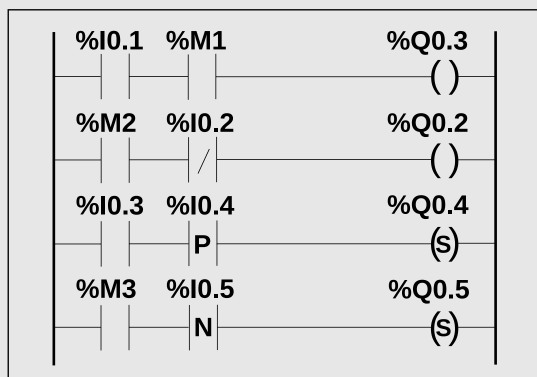

```
LD     %I0.1
ST     %Q0.3
STN    %Q0.2
S      %Q0.4

LD     %I0.2
R      %Q0.4
```

指令	操作数
ST	%Q,%M,%S,%BLK,x,%*:Xk
STN	%Q,%M,%S,%BLK,x,%*:Xk
S	%Q,%M,%S,%BLK.x,%*:Xk
LDF	%Q,%M,%S,%BLK.x,%*:Xk

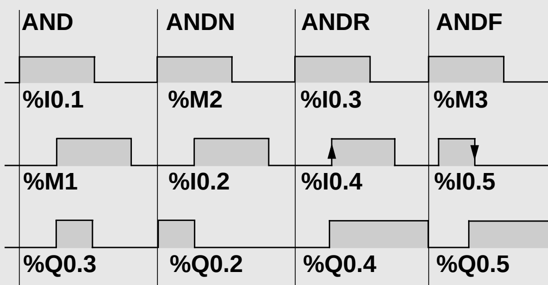
2.1-5 逻辑与指令：AND ANDN ANDR ANDF

这类指令在操作数（或它的取反数、上升沿和下降沿）和前面指令所产生的布尔结果之间进行逻辑与操作。



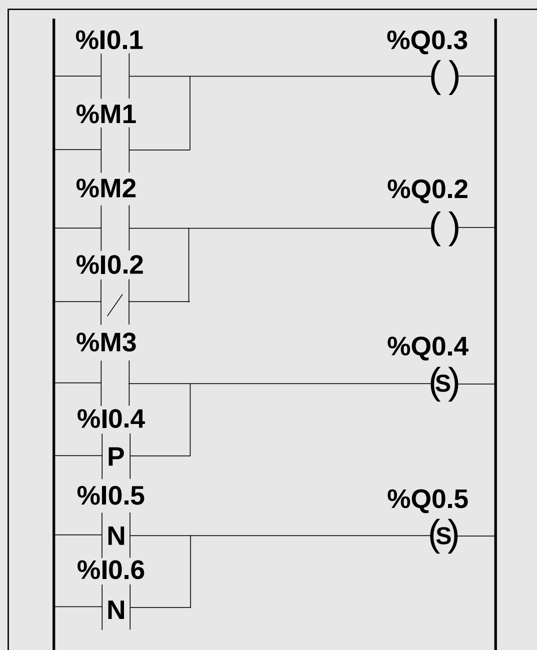

```
LD    %I0.1
AND   %M1
ST    %Q0.3
LD    %M2
ANDN  %I0.2
ST    %Q0.2
LD    %I0.3
ANDR  %I0.4
S     %Q0.4
LD    %M3
ANDF  %I0.5
S     %Q0.5
```

指令	操作数
AND	0/1,%I,%Q,%M,%S,%BLK,x,%*:Xk,[
ANDN	%I,%Q,%M,%S,%BLK.x,%*:Xk,[
ANDR	%I %M
ANDF	%I %M

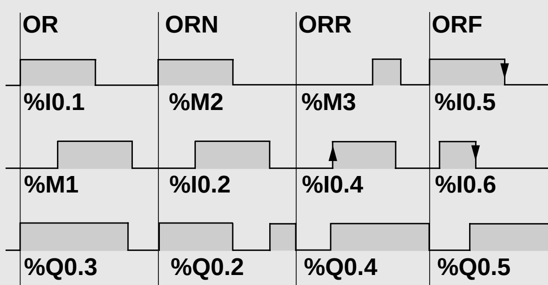


2.1-6 逻辑或指令：OR ORN ORR ORF

这条指令在操作数（或它的取反数、上升沿和下降沿）和前面指令所产生的布尔结果之间进行逻辑或操作。




```
LD    %I0.1
OR    %M1
ST    %Q0.3
LD    %M2
ORN   %I0.2
ST    %Q0.2
LD    %M3
ORR   %I0.4
S     %Q0.4
LDF   %I0.5
ORF   %I0.6
S     %Q0.5
```

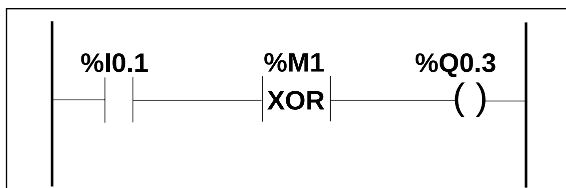


指令	操作数
OR	0/1,%I,%Q,%M,%S,%BLK,x,%*:Xk
ORN	%I,%Q,%M,%S,%BLK.x,%*:Xk
ORR	%I %M
ORF	%I %M

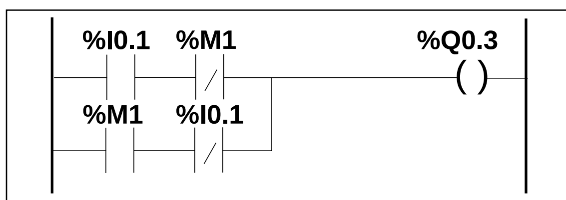
2.1-7 异或指令：XOR XORN XORR XORF

这条指令在操作数（或它的取反数、上升沿和下降沿）和前面指令所产生的布尔结果之间进行逻辑异或操作。

异或指令可由以下任意一种方式执行，例如：

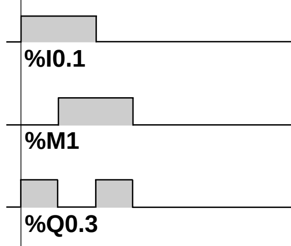



```
LD     %I0.1
XOR    %M1
ST     %Q0.3
```




```
LD     %I0.1
ANDN   %M1
OR(    %M1
ANDN   %I0.1
)
ST     %Q0.3
```

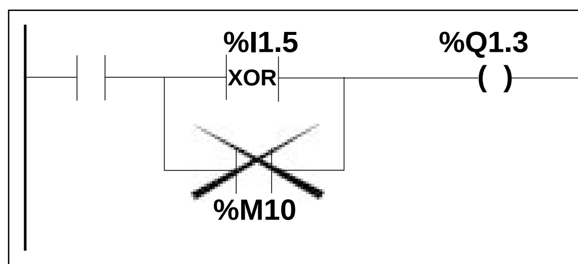
指令	操作数
XOR	%I,%Q,%M,%S,%BLK.x,%*:Xk
XORN	%I,%Q,%M,%S,%BLK.x,%*:Xk
XORR	%I %M
XORF	%I %M



特殊情况

- 在梯形图中异或线圈不能
- 位于梯级的左边（首位置），
- 并行放置

例如：试图加入下面的梯级会发生确认错误。



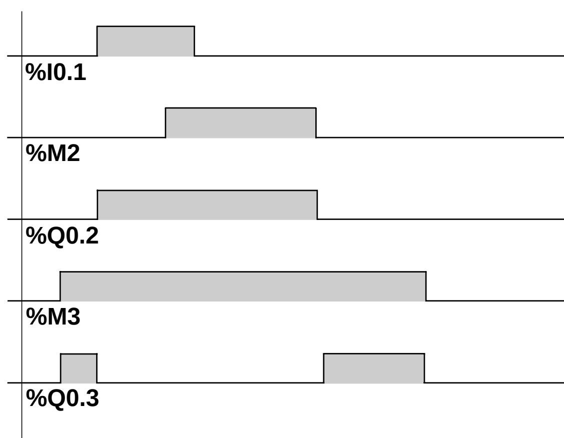
2.1-8 取非指令：N

这条指令将前面指令的执行结果取反。

```
LD    %I0.1
OR    %M2
ST    %Q0.2
N
AND   %M3
ST    %Q0.3
```

指令	操作数
N	–

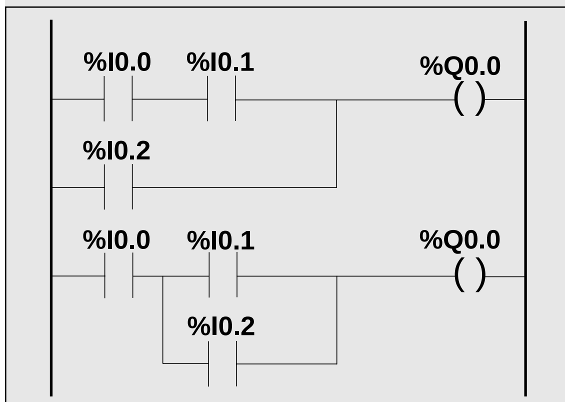
注意：指令 N 不可逆。



2.1-9 圆括号的使用

与指令和或指令可使用圆括号。圆括号可用于梯形图中的并列部分。左括号紧跟在与和或指令之后，右括号则表示执行括号内的所有内容。每个左括号必须有一个右括号与其相对应。

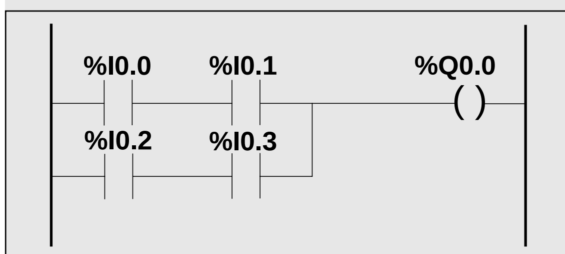
例如：AND(...)




```
LD      %I0.0
AND     %I0.1
OR      %I0.2
ST      %Q0.0

LD      %I0.0
AND(    %I0.1
OR      %I0.2
)
ST      %Q0.0
```

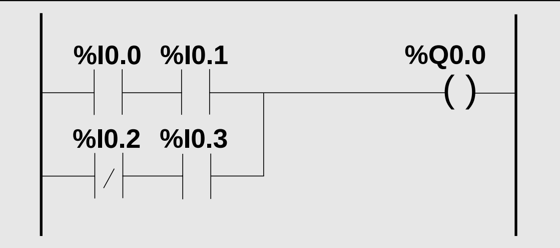
例如：OR(...)




```
LD      %I0.0
AND     %I0.1
OR(     %I0.2
AND     %I0.3
)
ST      %Q0.0
```

下列操作可以加上圆括号：N，F，R 或 [

- N　取反，例如 AND(N 或 OR(N
- R　上升沿，例如 AND(R 或 OR(R
- F　下降沿，例如 AND(F 或 OR(F
- [　比较，见 3.1-5 节

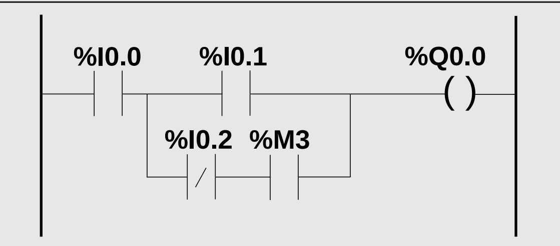



```
LD      %I0.0
AND     %I0.1
OR(N    %I0.2
AND     %I0.3
)
ST      %Q0.0
```

括号的嵌套

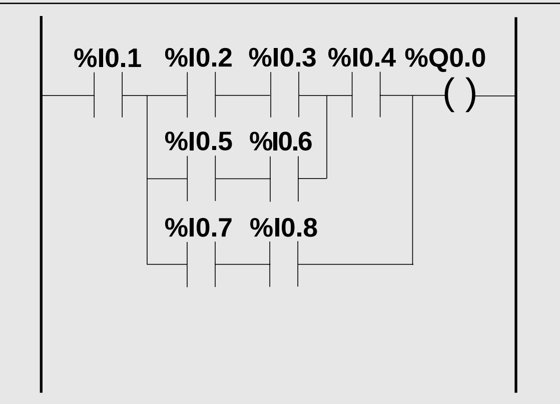
允许嵌套 8 层括号。

例如




```
LD      %I0.0
AND(    %I0.1
OR(N    %I0.2
AND     %M3
)
)
ST      %Q0.0
```

例如




```
LD      %I0.1
AND(    %I0.2
AND     %I0.3
OR(     %I0.5
AND     %I0.6
)
AND     %I0.4
OR(     %I0.7
AND     %I0.8
)
)
ST      %Q0.0
```

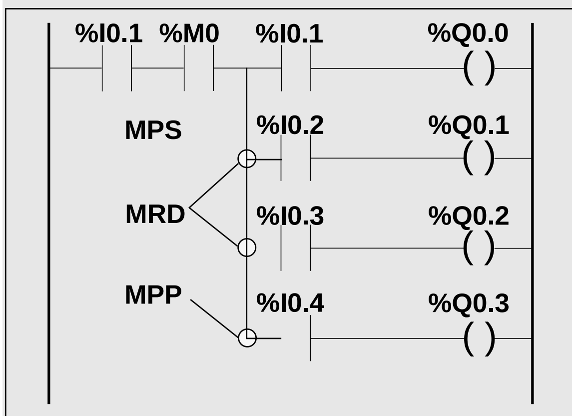
注意：

- 左右括号应对称。
- 标号 %Li：和 子程序 SRi：不可以加括号。类似的还有跳转指令 (JMP)，子程序指令 (SRi) 和功能块指令。详细说明见 2.4-3 节。
- 赋值指令 ST、STN、S 和 R 不可置于括号内。

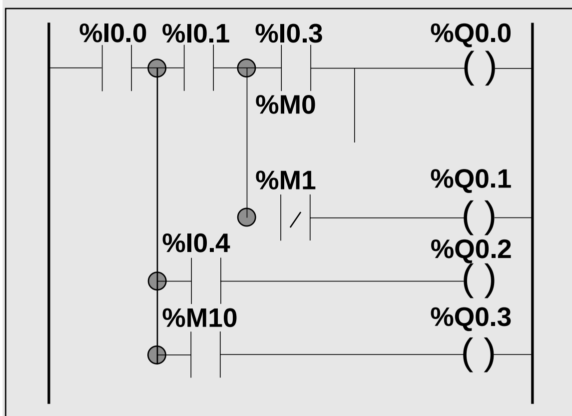
2.1-10 指令 MPS MRD MPP

这三条指令用于处理与线圈的连路。它们使用一个临时存储区作为存放最多 8 个布尔表达式的堆栈。指令 MPS 将累加器值（最近一次逻辑运算的结果）推入堆栈顶部，并使堆栈中的其他值向堆栈底部移动一格。指令 MRD 将堆栈顶部值读入累加器。指令 MPP 将堆栈顶部值读入累加器并将堆栈内其他值向顶部移动一格。

例如：




```
LD      %I0.0
AND     %M0
MPS
AND     %I0.1
ST      %Q0.0
MRD
AND     %I0.2
ST      %Q0.1
MRD
AND     %I0.3
ST      %Q0.2
MPP
AND     %I0.4
ST      %Q0.3
```



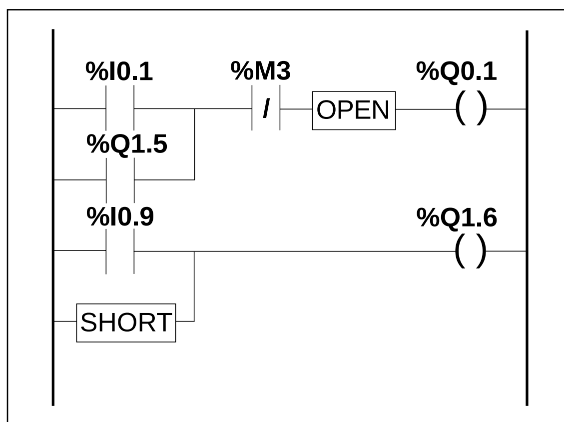

```
LD      %I0.0
MPS
AND     %I0.1
MPS
AND(    %I0.3
OR      %M0
)
ST      %Q0.0
MPP
ANDN    %M1
ST      %Q0.1
MRD
AND     %I0.4
ST      %Q0.2
MPP
AND     %M10
ST      %Q0.3
```

注意：以上指令不能用于括号内。

2.1-11 特殊梯形图指令 OPEN 和 SHORT

为了便于梯形图的编程，在调试和排除故障时可使用这两条指令。OPEN 和 SHORT 指令可以通过 “ 断路 ” 或 “ 短路 ” 来改变梯级的逻辑值。“ 断路 ” 是指不管梯级的逻辑值如何，断开它与后面梯级的连路； “ 短路 ” 是指不管梯级的逻辑值如何，连通它与后面梯级的连路。

其后紧跟立即值 1 和 0 的 OR 和 AND 指令可作为 OPEN 和 SHORT 指令, 如下面的程序。




```
LD      %I0.0
OR      %Q1.5
ANDN    %M3
AND     0
ST      %Q0.1
LD      %I0.9
OR      1
ST      %Q1.6
```

2.2 标准功能块

2.2-1 与标准功能块相关的位对象和字对象

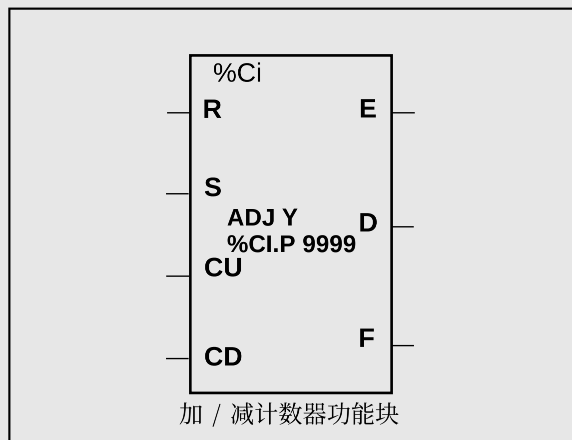
建立功能块位对象和特殊字。

● 位对象：

它们对应于功能块的输出，可由布尔测试指令进行访问。

它们有多种被寻址方式：

- 直接寻址（例如 LD E），条件是它们在可逆编程中与块相连。（见 2.2-2 节）
- 指定功能块类型（例如 LD %Ci.E），输入可以指令方式访问。



加 / 减计数器功能块

● 字对象：

它们对应于：

- 功能块配置参数，它们中有的能被程序访问（例如预设置参数），有的则不能（例如时基），
- 当前值（例如 %Ci.V 当前计数值）。

程序可访问的功能块字对象和位对象列表

标准功能块	对应字和位		地址	写访问	见节
定时器 %TMi（i=0-31）	字	当前值	%TMi.V	不能	2.2-3
		预设值	%TMi.P	能	
	位	定时器输出	%TMi.Q	不能	
加／减 计数器	字	当前值	%Ci.V	不能	2.2-4
		预设值	%Ci.P	能	
%Ci（i=0 to-15）	位	下溢输出（空）	%Ci.E	不能	
		预设达到输出	%Ci.D	不能	
		满溢输出（满）	%Ci.F	不能	
LIFO/FIFO 寄存器 %Ri（i= 0-3）	字	寄存器输入	%Ri.I	能	2.2-5
		寄存器输出	%Ri.O	能	
	位	寄存器满输出	%Ri.F	不能	
		寄存器空输出	%Ri.E	不能	
鼓形控制器	字	当前步号	%DRi.S	能	2.2-6
%DRi（i=0-3）	位	当前步为最后一步	%DRi.F	不能	

2.2-2 编程原则

标准功能块有两种编程方式：

- 使用标准功能块指令（例如 BLK %TM2）：梯形图语言中这种可逆的编程方法使得运算可在程序的功能块中执行。
- 使用特殊指令（例如 CU %Ci）：这种不可逆的方法运算要在程序的几个块输入处执行（例如 行100 CU %C1，行 174 CD %C1，行 209 LD %C1.D）。

标准功能块可逆编程的原则

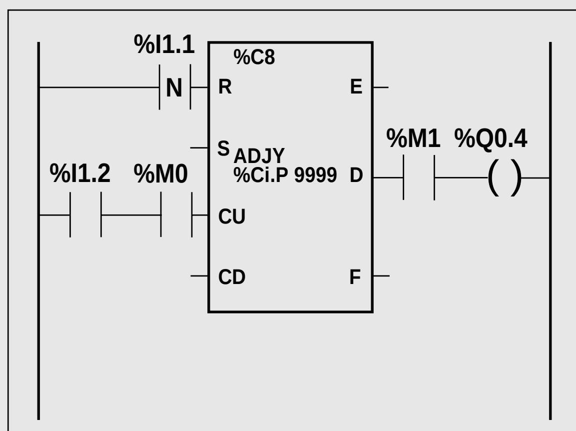
这种编程使用指令 BLK，OUT_BLK 和 END_BLK.

BLK 表示功能块的开始

OUT_BLK 表示接通功能块的输出

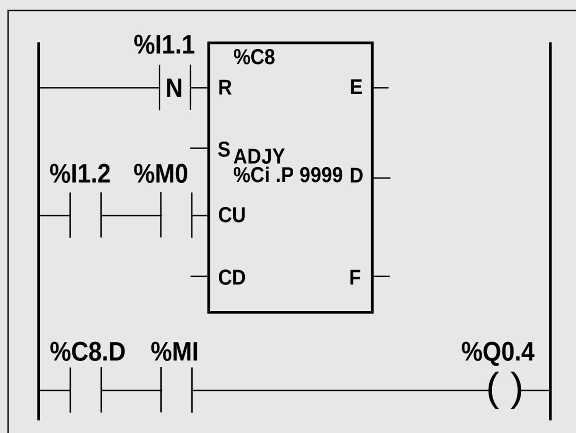
END_BLK 表示功能块的结束

带有输出的可逆编程举例



BLK	%C8	输入操作
LDF	%I1.1	
R		
LD	%I1.2	
AND	%M0	
CU		
OUT_BLK		输出操作
LD	D	
AND	%M1	
ST	%Q0.4	
END_BLK		

没有输出的可逆编程举例

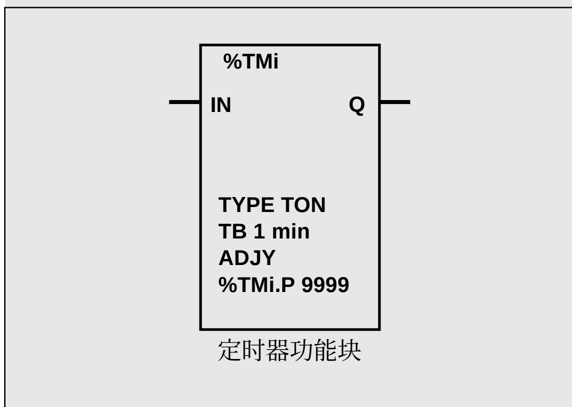


BLK	%C8	输入操作
LDF	%I1.1	
R		
LD	%I1.2	
AND	%M0	
CU		
END_BLK		输出操作
LD	%C8.D	
AND	%M1	
ST	%Q0.4	

注意：

只有相应功能块中的测试和输入指令可以放在指令 BLK 和 OUT_BLK 之间（如果程序中没有 OUT_BLK 就放在 BLK 和 END_BLK 之间）。

2.2-3 定时器功能块 %TMi



定时器功能块

有三种定时器：

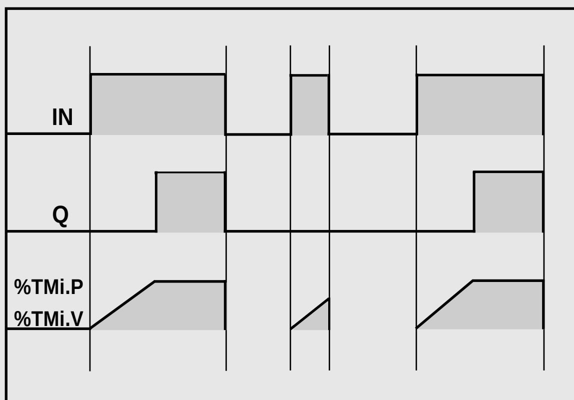
- **TON**：这种定时器用于控制导通－延时动作。这种延时是可编程的并可由编程终端进行修改。
- **TOF**：这种定时器用于控制关断－延时动作。这种延时是可编程的并可由编程终端进行修改。
- **TP**： 这种定时器用于产生精确宽度的脉冲。脉冲宽度是可编程的并可根据编程终端进行修改。

注意：预设值越大，定时器的相对精确度越高。

特性		
定时器编号	%TMi	0 到 31
类型	TON TOF TP	• 导通－延时（默认值） • 关断－延时 • 脉冲（单稳态）
时基	TB	1 分钟（默认值）、1 秒、100 毫秒，10 毫秒，1 毫秒（用于 TM0 和 TM1）
预设值	%TMi.P	0-%TMi.P-9999．可由程序读取、测试和写入，默认值为 9999。其长度和延迟应等于 %TMi.P x TB.
数据编辑器	Y/N	Y:可以在数据编辑器中改变预设值 %TMi.P。 N:在数据编辑器中不能访问。
设置输入（或指令）	IN	定时器从上升沿（TON 或 TP 类型）或下降沿（TOF)类型开始。
定时器输出	Q	根据实现的功能：TON、TOF 或 TP，相关位 %TMi.Q 置 1。

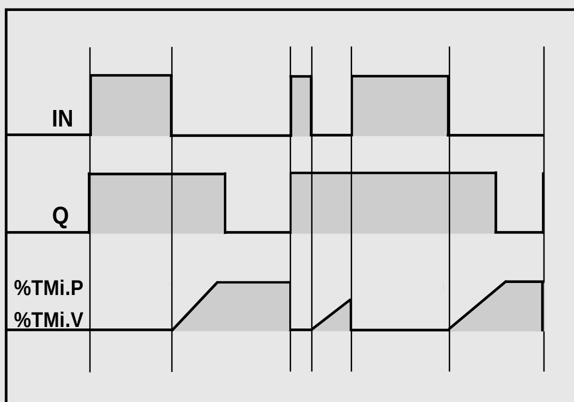
(1)根据数据编辑器终端修改 %TMi.V

用作导通－延迟定时器:**TON类型**



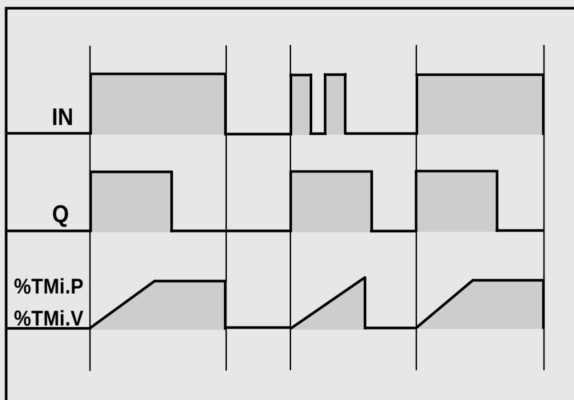
定时器在输入 IN (1) 的上升沿开始工作：当前值 %TMi.V 以时基 TB 为单位从 0 增加到 %TMi.P。当当前值达到 %TMi.P 时，输出位 %TMi.Q 变为 1。当输入 IN (2) 被观测到下降沿，即使定时器还未达到 % TMi.P，定时器也要停止工作。

用作关断－延迟定时器：**TOF** 类型



在遇到输入 IN (1) 的上升沿时, 当前值 %TMi.V 置为 0 （即使定时器仍在工作)。当输入 IN 被观测到下降沿，定时器开始工作。当前值以时基 TB 为单位从 0 增加到 %TMi.P。当输入 IN 被观测到一个上升沿时，输出位 %TMi.Q 变为 1，当当前值达到 %TMi.P 时，定时器变回为 0。

用作脉冲：**TP** 类型

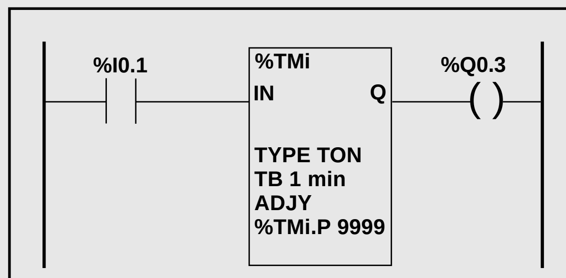


在输入 IN (1) 的上升沿定时器开始工作：当定时器开始工作时，输出位 %TMi.Q 变为 1，当前值 %TMi.V 从 0 开始以时基 TB 为单位增加至 %TMi.P，当当前值达到 %TMi.P 时，输出位 %TMi.Q 变回为 0。
这个定时器不可复位。 如果 %TMi.V 等于 %TMi.P，且输入 IN 的状态为 0，则 %TMi.V 置为 0。

(1) 指令 IN 被激活
(2) 指令 IN 被检测到

编程和配置

不论定时器的功能块用途如何，它们的编程方法相同。配置时选择功能 TON、TOF 和 TP。



可逆编程

```
BLK     %TM1
LD      %I0.1
IN
OUT_BLK
LD      Q
ST      %Q0.3
END_BLK
```

配置

在配置时必须加入以下参数。

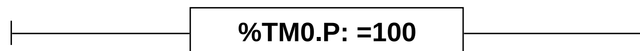
- 类型：TON、TOF 或 TP
- TB：1 分钟、1 秒 、100 毫秒、10 毫秒或 1 毫秒
- %TMi.P：0 到 9999
- 调节：Y 到 N

不可逆编程

```
LD    %I0.1
IN    %TM1
LD    %TM1.Q
ST    %Q0.3
```

在程序中修改定时器设定值

在程序中只需对 %TMi.P 进行赋值，即可修改定时器的设定值，如：



程序例是将定时器 0 的设定值设为 100xTB

特殊情况

- 冷启动的影响：(%S0=1)
- 强置当前值为 0,
- 置输出 %TMi.Q 为 0,
- 预设值被重设为配置过程中定义的值。
- 热启动的影响： (%S1=1) 对定时器的当前值和预设置都没有影响。当电源损耗时当前值不变。
- **PLC** 停止工作的影响：停止 PLC 不会停止定时器，定时器当前值会继续增加。
- 程序跳转的影响：跳过一个定时器块并不停止定时器。定时器会持续增加直到达到预设值 (%TMi.P)。此时，定时器块输出位 Q (%TMi.Q)的状态改变； 然而，与块输出直接相连的输出侧没有被激活，不能被 PLC 扫描到。
- 位 **%TMi.Q**（完成位）测试：程序中最好只进行一次位 %TMi.Q 测试。
- 修改预设值 **%TMi.P** 的影响：只有当定时器重新被激活时，根据指令改变或调整预设值才会起作用。

时基为 **1** 毫秒的定时器

1 毫秒时基只适用于定时器 %TM0 和 %TM1。

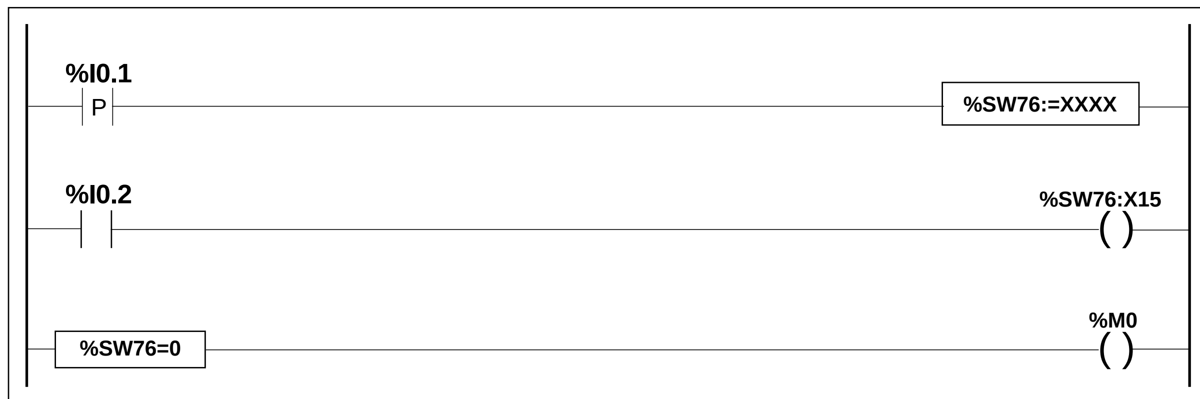
如果用户需要，可使用四个系统字 %SW76、%SW77、%SW78 和 %SW79 作为 " 时钟影象 "。

由程序或编程终端给这四个系统字赋予一个正值，则它们以毫秒为单位进行递减，赋予的这个正值即为定时器预设值。

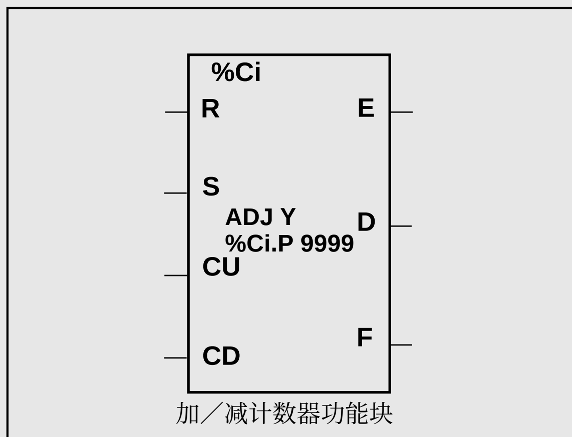
可以通过将相应的系统字中的 0-15 位中的 1 个位置 1，来停止定时，复位为 0 来允许启动定时器。

编程举例

```
LDR%I0.1            （在 %I0.1 的上升沿启动定时器）
[%SW76:=XXXX]       （XXXX= 预设值）
LD %I0.2            （输入 I0.2，停止该定时器）
ST %SW76:X15
LD [%SW76=0]        （定时器结束测试）
ST %M0
..........
```



2.2-4 加／减计数器功能块 %Ci



加／减计数器功能块

加／减计数器功能块用于加／减记录事件数。这两种运算可以同时进行。

特性		
计数器编号	**%Ci**	0 至 15
当前值	**%Ci.V**	字根据输入（或指令）CU 和 CD 进行增加或减少。可以用程序（1）读取或测试，但不可以写入。
预设值	**%Ci.P**	0-%Ci.P-9999.字可以读取、测试和写入(2)（默认值：9999)。
根据终端编辑	**Y/N**	Y：可在数据编辑器中修改预设值 N：不可访问数据编辑器。
复位输入（或指令）	**R**	状态为1：%Ci.V = 0
设置输入（或指令）	**S**	状态为1： %Ci.V = %Ci.P.
加运算输入（或指令）	**CU**	在上升沿处增加 %Ci.V 。
减运算输入（或指令）	**CD**	在上升沿处减少 %Ci.V 。
下溢输出	**E**（空值）	相应位 %Ci.E=1，当减计数器 %Ci.V 从 0 改变至 9999 （当 %Ci.V 到达 9999 后置为 1，如果计数器继续减少则复位为 0)。
预设输出	**D**（完成）	当 %Ci.V=%Ci.P 时，相应位 %Ci.D=1。
上溢输出	**F**（满溢）	当 %Ci.V 从 9999 变到 0 时（当 %Ci.V 达到 0 时置 1 ，如果计数器继续增加则复位为 0 。)，相应位 %Ci.F =1。

(1)可以根据数据编辑器的终端对 %Ci.V 进行修改。
(2)可以在程序中对 %Ci.P 进行赋值，修改计数器设定值。

操作

• 加计数：当加计数中的输入 CU 出现一个上升沿（或指令 CU 被激活），当前值 %Ci.V 加 1。当这个值等于预设值 %Ci.P，输出 D 输出，位 %Ci.D 变为 1。当 %Ci.V 从 9999 变为 0，输出位 %Ci.F 状态变为 1，如果计数器继续增加则状态复位为 0。

• 减计数：当减计数中的输入 CD 出现一个上升沿（或指令 CD 被激活），当前值 %Ci.V 减 1。当 %Ci.V 从 0 变为 9999，输出位 %Ci.E（减计数溢出）状态变为 1，如果计数器继续减少则状态复位为 0。

• 加／减计数：要同时使用加计数和减计数功能（或者激活指令 CD 和 CU），则必须对这 2 个相应的输入 CU 和 CD 加以控制。连续扫描这两个输入，如果它们都为 1，当前值保留不变。

• 复位：当输入置为 1（或者指令被激活），当前值 %Ci.V 被强置为 0，输出 %Ci.E、 %Ci.D 和 %Ci.F 则置为 0，并且复位输入优先。

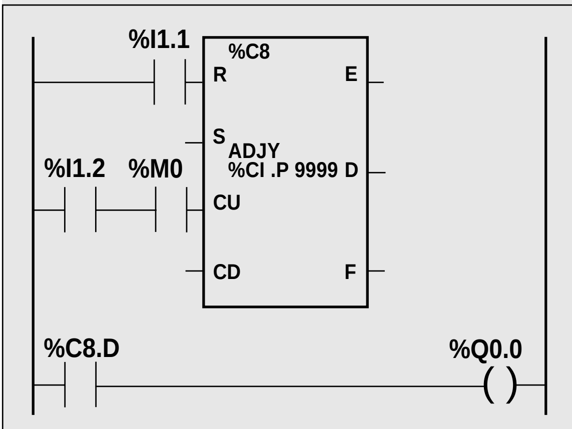
• 置值：如果输入 S 的状态为 1 （或者指令 S 激活）而且复位输入状态为 0 （指令 R 未激活），当前值 %Ci.V 取 %Ci.P 的值，而且输出 %Ci.D 置为 1。

特殊情况

- 冷启动的影响：**(%S0=1)**
  - 当前值 %Ci.V 置为 0。
  - 输出位 %Ci.E、%Ci.D 和 %Ci.F 都置为 0。
  - 根据配置初始化预设值。
- **PLC** 停止工作后，热启动的影响 **(%S1=1):** 对计数器（%Ci.V）的当前值没有影响。
- 修改预设值 **%Ci.P** 的影响：当功能块在应用时（有输出被激活），根据指令进行的预设值修改和调整会产生影响。

配置和编程

例如 : 对项目计数直至 5000。输入 %I1.2 的每个脉冲 (当内部位 %M0 为 1 时) 都使加计数器 %C8 增加，直至达到它的预设值 (位 %C8.D=1)。计数器的值由输入 %I1.1 复位。



配置

配置时必须加入下列参数 :

- %Ci.P，在该例中设为 5000
- 可调整: 是

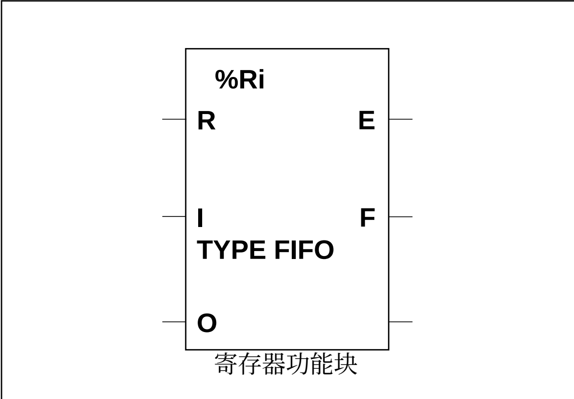
可逆编程

```
BLK     %C8
LD      %I1.1
R
LD      %I1.2
AND     %M0
CU
END_BLK
LD      %C8.D
ST      %Q0.0
```

不可逆编程

```
LD      %I1.1
R       %C8
LD      %I1.2
AND     %M0
CU      %C8
LD      %C8.D
ST      %Q0.0
```

2.2-5 LIFO/FIFO 寄存器功能块 %Ri



寄存器功能块

寄存器是一个存储 16 个 16 位字的内存块，有两种存储方式：

- 队列式（先进先出），如前文提及的 FIFO
- 堆栈式（后进先出），如前文提及的 LIFO

特性		
寄存器编号	%Ri	0 到 3
类型	FIFO	队列式（默认值）
	LIFO	堆栈式
输入字	%Ri.I	寄存器输入字，可读取、测试和写入。
输出字	%Ri.O	寄存器输出字，可读取、测试和写入。
存储输入（或指令）	I (In)	在上升沿处，将字 %Ri.I 的值存入寄存器。
取出输入（或指令）	O (Out)	在上升沿处，将一个数据字装入字 %Ri.O 内。
复位输入（或指令）	R (Reset)	%Ri.O 状态为 1 时，初始化寄存器。
空输出	E (Empty)	对应位 %Ri.E 表示寄存器为空。可测试。
满输出	F (Full)	对应位 %Ri.F 表示寄存器为满。可测试。

操作

FIFO（先进先出）

最先进入的数据项最先被取出。

当接到一个存储请求（输入 I 处的上升沿或者指令 I 被激活），输入字 %Ri.I 的值（已经被装入）被存放到队列的顶端（图 a）。

当队列已满时（输出 F=1），不可以再存入数据。

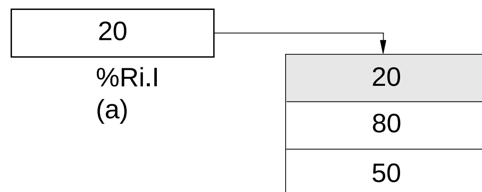
当接到一个取出请求（输入 O 处的上升沿或者指令 O 被激活），队列最底部的数据字被装入输出字 %Ri.O，并且寄存器队列中的数据都往底部移一格（图 b）。

当寄存器是空的时候（输出 E=1），不能再取出数据。输出字 %Ri.O 不变，保持原值。

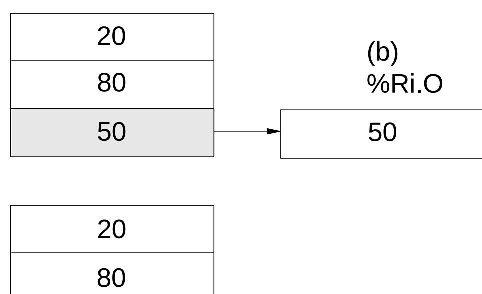
可以在任何时候复位寄存器（输入 R 为 1 或者指令 R 被激活）。

例如：

把 %Ri.I 的值存入队列的顶端。



取出第一个数据项装入 %Ri.O。



LIFO（后进先出）

最后进入的数据项最先被取出。

当接到一个存储请求（输入 I 处的上升沿或者指令 I 被激活），输入字 %Ri.I 的值（已经被装入）被存放到堆栈的顶端（图 c）。

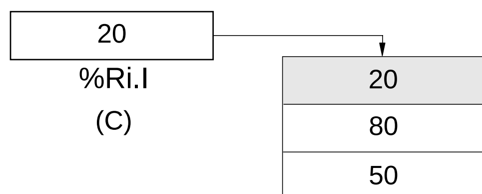
当堆栈已满时（输出 F=1），不可以再存入数据。

当接到一个取出请求（输入 O 处的上升沿或者指令 O 被激活），堆栈顶端的数据（最后进入的数据）被装入输出位 %Ri.O（图 d）。

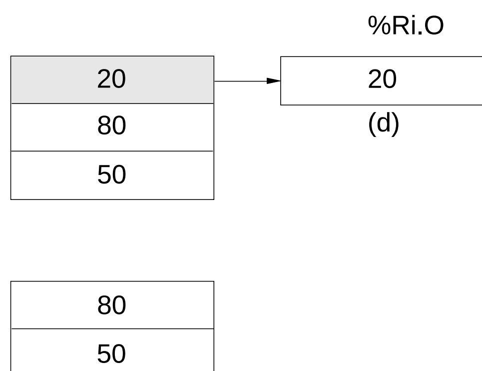
当寄存器是空的时候（输出 E=1），不能再取出数据。输出字 %Ri.O 不变，保持原值。可以在任何时候复位寄存器（输入 R 为 1 或者指令 R 被激活）。指针指向堆栈的顶端。

例如：

把 %Ri.I 的值存入堆栈的顶端。



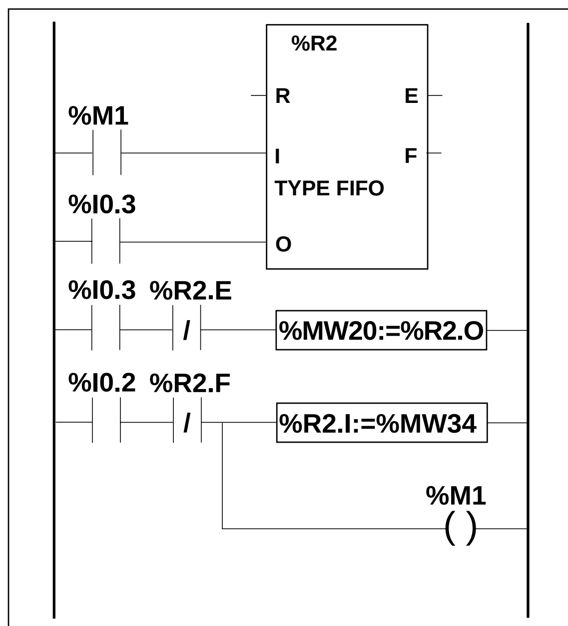
把堆栈顶端的数据项取出装入 %Ri.O。



特殊情况

- 冷启动的影响 :(%S0=1) 初始化寄存器输出位 %Ri.E，设输出 E 为 1， 并把字 %Ri.I 和 %Ri.O 复位为 0。
- 热启动的影响 :(%S1=1) 对寄存器的值和输出位的状态没有影响。

编程和配置



该例演示了寄存器 R2 未满时 (%R2.F=0)，使用存储请求 %I0.2 把字 %MW34 的内容装入 %R2.I。存储请求由 %M1 来实现。取出请求由输入 %I0.3 来实现，此时若寄存器不空 (%R2.E=0)，则 %R2.O 的内容被装入 %MW20。

配置

配置时必须加入寄存器类型这个参数：FIFO（默认值）或 LIFO。

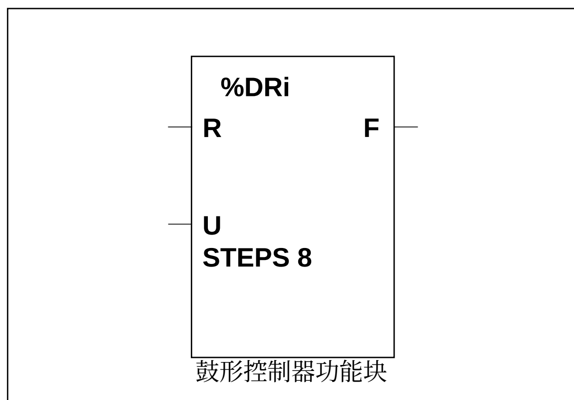
可逆程序

```
BLK     %R2
LD      %M1
I
LD      %I0.3
O
END_BLK
LD      %I0.3
ANDN    %R2.E
[%MW20:=%R2.O]
LD      %I0.2
ANDN    %R2.F
[%R2.I:=%MW34]
ST      %M1
```

不可逆程序

```
LD      %M1
I       %R2
LD      %I0.3
O       %R2
LD      %R2.E
[%MW20:=%R2.O]
LD      %I0.2
ANDN    %R2.F
[%R2.I:=%MW34]
ST      %M1
```

2.2-6 鼓形控制器功能块 %DRi



鼓形控制器功能块

鼓形控制器功能块的工作原理与机电类电子凸轮器相似，也是根据外部环境改变步序。机电类电子凸轮器的控制器中凸轮的高点给出的命令由该控制器执行。相应的，在鼓形控制器功能块中，用状态为 1 来代表每一步的高点，并赋值给输出位 %Qi.j 或内部位 %Mi 作为控制位。

特性		
编号	%DRi	0 到 3
当前步号	%DRi.S	0-%DRi.S-7 可被读取和测试。 只能以十进制数的格式在程序中写。
步数		1 到 8（默认值）
回到 0 步输入（或指令）	R (RESET)	状态为 1 时，将鼓形控制器块置 0 步。
前进输入（或指令）	U(UP)	在上升沿处，使鼓形控制块向前进一步并更新控制位。
输出	F (FULL)	表示当前步等于最后一步。可测试对应位 %DRi.F。
控制位		与步（16 位控制位）对应的输出或内部位，在配置编辑器内加以定义。

操作

鼓形控制器包括：

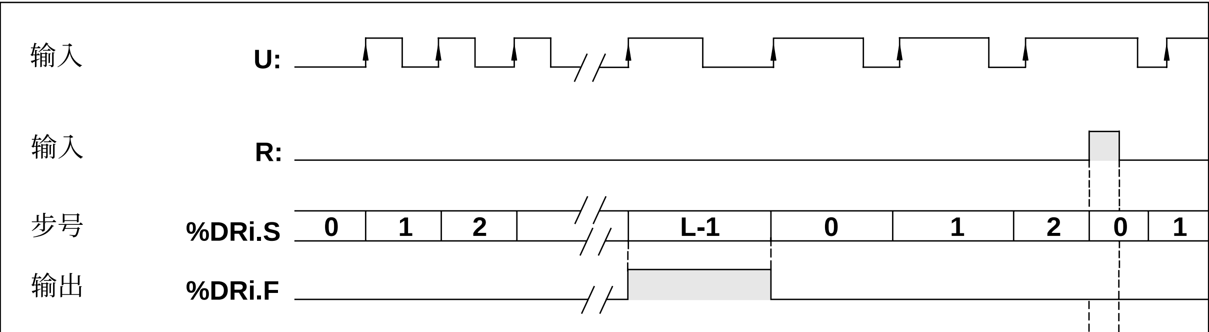
- 由 8 步（0 到 7）16 个数据位（每步的状态）组成的常数（凸轮）矩阵，其中数据位被排放在 0 到 F 个竖列中。
- 控制位表（每列一个），相应的可以是输出 %Qi.j ,也可以是内部位 %Mi。在当前步中，控制位以二进制表示该步的状态。

下表概括了鼓形控制器块的主要特性

控制位	0 %Q0.1	1 %Q0.3	2 %Q1.5		D %Q0.6	E %Q0.5	F %Q1.0
步 **0**	**0**	**0**	**1**		**1**	**1**	**0**
步 **1**	**1**	**0**	**1**		**1**	**0**	**0**
¦							
步 **5**	**1**	**1**	**1**		**0**	**0**	**0**
步 **6**	**0**	**1**	**1**		**0**	**1**	**0**
步 **7**	**1**	**1**	**1**		**1**	**0**	**0**

在上例中, 5 号步是当前步, 把控制位 %Q0.1；%Q0.3 和 %Q1.5 置为 1；把控制位 %Q0.6、%Q0.5 和 %Q1.0 置为 0。
在输入 U 的每个上升沿处（或执行指令 U 时），当前步向前进。当前步可由程序修改。

运行图

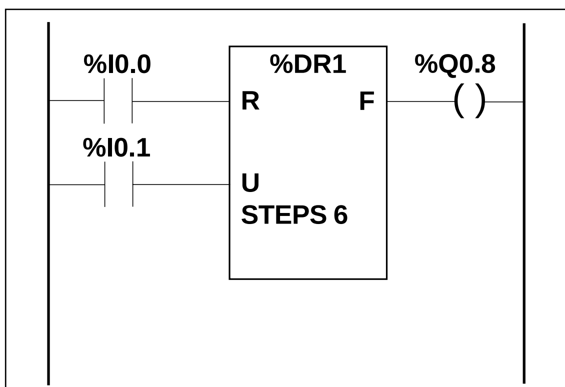


特殊情况

- **冷启动的影响:** (%S0=1) 将鼓形控制器功能块复位为步 0 并刷新控制位。
- **热启动的影响:** (%S1=1) 过了当前步后刷新控制位。
- **程序跳转的影响:** 如果没有扫描到鼓形控制器, 控制位不会复位为 0。
- **更新控制位:** 只发生在步变化或冷热启动时。

编程和配置

在这个例子中，由输入 %I0.1 控制， 每次置 1，前 6 个输出 %Q0.0 到 %Q0.5 被依次执行。输入 %I0.0 将输出复位为 0。



可逆编程

```
BLK       %DR1
LD        %I0.0
R
LD        %I0.1
U
OUT_BLK
LD         F
ST         %Q0.8
END_BLK
```

配置

配置时作如下定义：

- 步数： 6
- 鼓形控制器每步的输出状态（控制位）

Q 0. 0: 1: 2: 3: 4: 5:

步 0: 0 0 0 0 0 0

步 1: 0 1 0 0 0 0

步 2: 0 0 1 0 0 0

步 3: 0 0 0 1 0 0

步 4: 0 0 0 0 1 0

步 5: 0 0 0 0 0 1

不可逆编程

```
LD        %I0.0
R         %DR1
LD        %I0.1
U         %DR1
LD        %DR1.F
ST        %Q0.8
```

- 控制位的分配

  0:%Q0.0 1:%Q0.1

  2:%Q0.2 3:%Q0.3

  4:%Q0.4 5:%Q0.5

2.3 程序指令集

2.3-1 程序结束指令 END ENDC ENDCN

用指令 END、ENDC 和 ENDCN 来定义程序的结束：

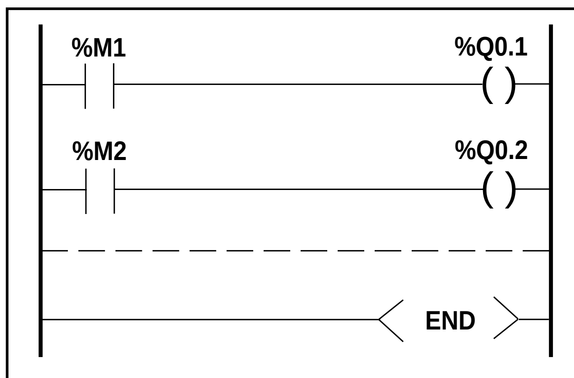
END：无条件程序结束

ENDC：如果前一个测试指令的布尔值为 1,则程序结束

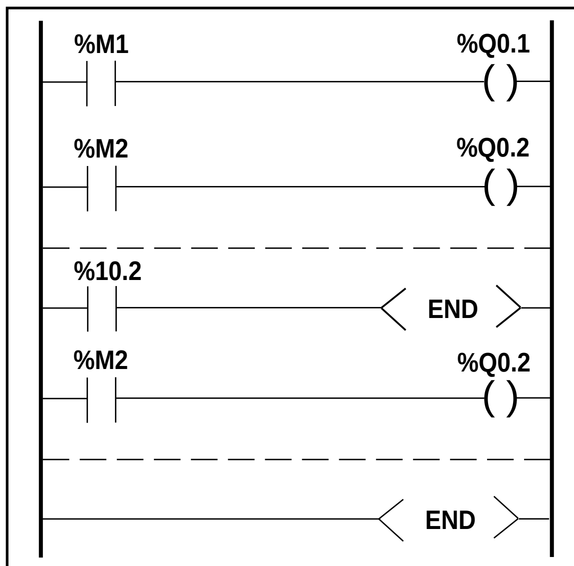
ENDCN：如果前一个测试指令的布尔值为 0,则程序结束

在常规扫描模式下，当执行结束指令时，输出刷新，并开始下一个扫描。
如果扫描是周期性的，一个周期结束时输出被刷新，并开始下一个周期。

例如：




```
LD      %M1
ST      %Q0.1
LD      %M2
ST      %Q0.2
.....................
END
```




```
LD      %M1
ST      %Q0.1
LD      %M2
ST      %Q0.2
.....................
LD      %I0.2
ENDC           → 如果 %I0.2=1,
LD      %M2      结束程序扫描
ST      %Q0.2
.....................  如果%I0.2=0, 继续程
END                    序扫描直至新的 END
                       指令
```

2.3-2 NOP 指令

NOP 指令不进行任何操作。它用来在程序中“ 保留 ”行，以便用户以后插入指令时无需修改行号。

2.3-3 跳转指令 **JMP JMPC JMPCN**，跳至标号为 **%Li:** 的程序行:

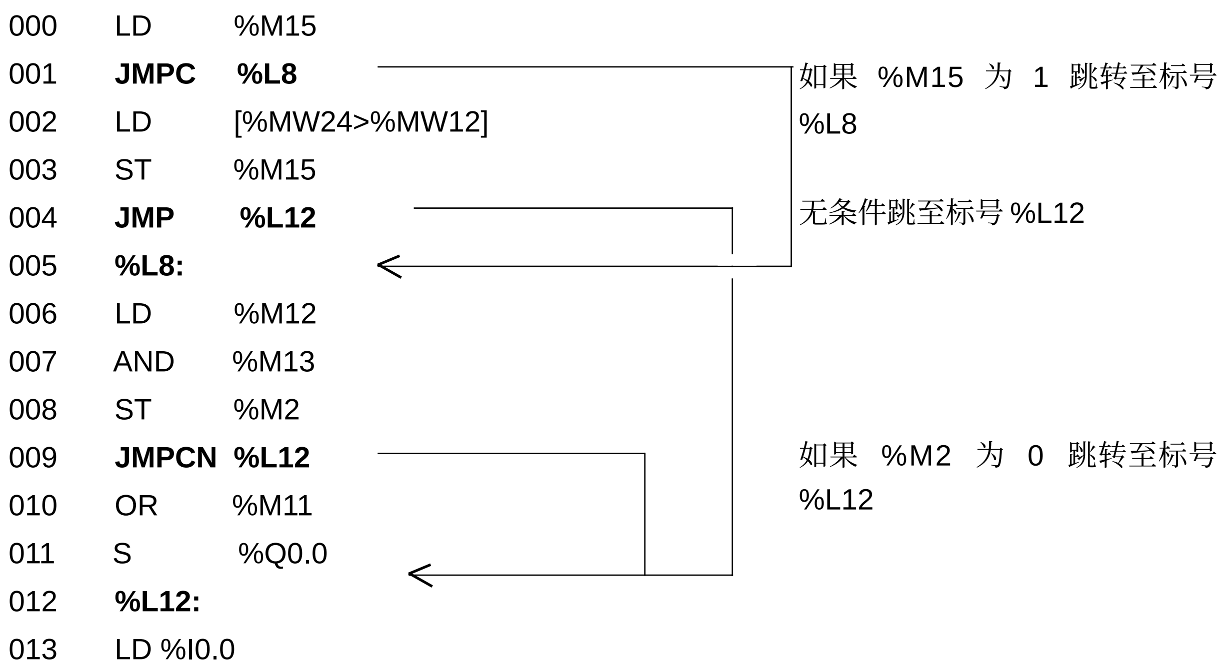
指令 JMP、JMPC 和 JMPCN 能立即中断执行, 转入执行标号为 %Li (i = 0 到 15)的程序行。

JMP:无条件程序跳转

JMPC:如果前一个指令的布尔值为 1，则程序跳转

JMPCN:如果前一个指令的布尔值为 0，则程序跳转

例如：



```
000   LD      %M15
001   JMPC    %L8
002   LD      [%MW24>%MW12]
003   ST      %M15
004   JMP     %L12
005   %L8:
006   LD      %M12
007   AND     %M13
008   ST      %M2
009   JMPCN   %L12
010   OR      %M11
011   S       %Q0.0
012   %L12:
013   LD %I0.0
```


注意：

- 这类指令不可用于括号内。
- 标号只能放在指令 LD、LDN、LDR、LDF 或 BLK 之前。
- 标号 %Li 在程序中只能定义一次。
- 程序跳转可以向上也可以向下。当向上跳转时，应注意程序的扫描时间。延长扫描时间可能导致警戒时钟超时而停止 PLC 运行。

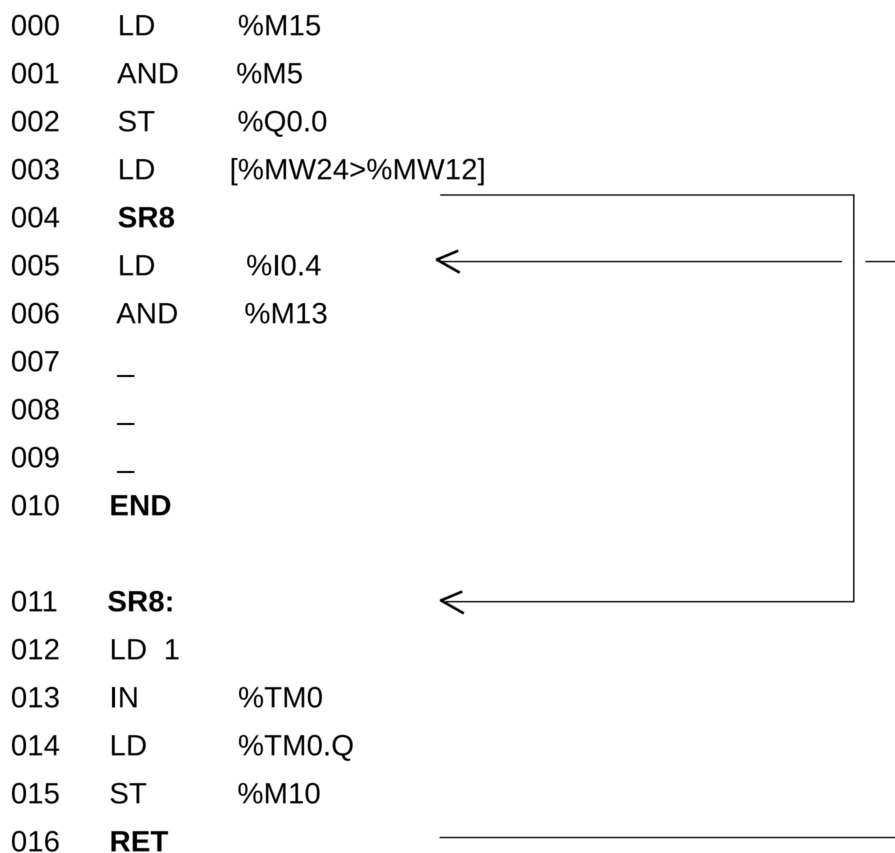
2.3-4 子程序指令 **SRn, SRn: RET**

SRn 指令调用标号为 SRn：的子程序。如果前一个布尔指令结果为 1。

RET 指令放在子程序的最后，用于返回主程序。

子程序的标号为 SRn: =0 到 15。

例如：



```
000   LD    %M15
001   AND   %M5
002   ST    %Q0.0
003   LD    [%MW24>%MW12]
004   SR8
005   LD    %I0.4
006   AND   %M13
007   _
008   _
009   _
010   END

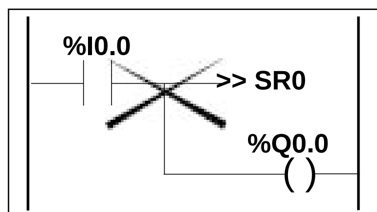
011   SR8:
012   LD 1
013   IN    %TM0
014   LD    %TM0.Q
015   ST    %M10
016   RET
```

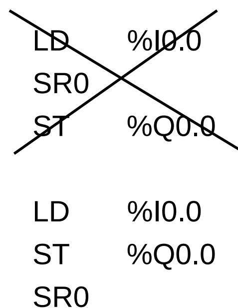
…………………………………

注意：

- 一个子程序不可以调用另一个子程序。
- 这类指令不可用于括号内。
- 标号只能放在指令 LD 或 BLK 之前，用于标志一个布尔等式或梯级的开始。
- 可在赋值指令（例如，输出或转移）之后调用子程序。

例如





```
LD    %I0.0
SR0
ST    %Q0.0

LD    %I0.0
ST    %Q0.0
SR0
```

3 数字指令和特殊指令

3.1 数字处理

3.1-1 字对象定义

字对象是存放在数据存储区中的 16 位字，它们可表示 -32768 到 32767 之间的任何整数（除了高速计数器是 0 到 65535）。

立即数

立即数是指与 16 位字同样格式的整数值，它将值赋于这些字，它们存放在程序内存中，表示 -32768 到 32767 之间的整数值。

字结构

字的内容或值根据下述约定以 16 位二进制码（或补码）的形式存放在用户内存中：

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	位的位置
0	1	0	1	0	0	1	0	0	1	0	0	1	1	0	1	位的状态
+ –	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	位的值

在带符号的二进制码中，第 15 位用于根据约定标示值的正负：

- 第 15 位为 0：字的值为正。
- 第 15 位为 1：字的值为负（负值用二进制补码逻辑表示）。

字和立即值可以以下形式存储和读取：

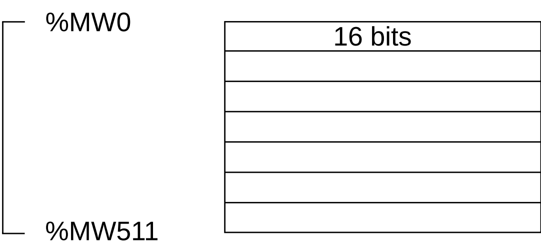
- 十进制　　1579 （最大值：32767，最小值：-32768）
- 十六进制　　16#A536 （最大值：16#FFFF，最小值：16#0000）

内部字

内部字用于存放程序运行时产生的值。

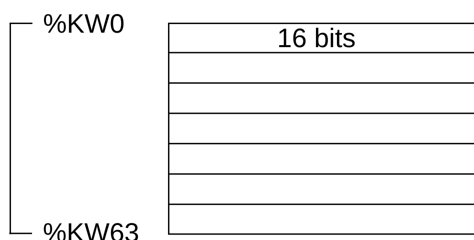
它们被存储在数据区中。

字 %MW0 到 %MW511 被用作工作字，可由程序直接读取或写入。



常量字

常量字是指常数值或者字母和数字报文。
它们只能由终端（在配置编辑器中）进行写入或修改。
程序对常量字 %KW0 - %KW63 只有读访问权。



I/O 字

I/O 字指输入字 %Wi,j 和输出字 %QWi,j。
它们用于对等 PLC 之间的数据交换或用来存储模拟量单元的 AD、 DA 数据，见节 3.5。

系统字

这类 16 位字有多种功能。读字 %SWi 可以访问直接来自 PLC 的数据，用于在应用程序中实现具体操作（例如，调节调度模块）。
每个系统字的作用请参见节 6。

字的位对象

可以从某些字的 16 位中取出一位，这一位可用冒号隔开，并被加入到字地址中。

语法结构：% 字对象：Xk，其中 k 表示对象字 0 到 15 位中的一位。
例如：%MW5:x6: 内部字 %MW5 中的第 6 位。

字操作数列表

类型	地址（或值）	最大个数	写访问	参考节
立即数 • 十进制 • 十六进制	例如：2103 例如：16#AF0D		不可以	
内部字	%MWi	512	可以	_
常量字	%KWi	64	不可以 (1)	_
系统字	%SWi	128	根据 i	5.2
功能块字	%TMi.P %Ci.P 等			2.2-1
I/O 字 • 输入字 • 输出字	%IWi.j %QWi.j	20 10	不可以 可以	3.3
字抽取位 • 内部	%MWi:Xk	512x 16	可以	
• 系统	%SWi:Xk	128 x 16	根据 i	
• 常量	%KWi:Xk	64 x 16	不可以	
• 输入	%IWi.j:Xk	20 x 16	不可以	
• 输出	%QWi.j:Xk	10 x 16	可以	

(1) 常量字应在配置模式中写入。

B

3.1-2 结构化对象

位串

位串是指一系列类型相同的相邻对象位， 并被定义为长度：L 。

位串举例：

%M8:6 (1)

%M8	%M9	%M10	%M11	%M12	%M13

类型	地址	最大范围	写访问
开关量输入位	%I0:L 或 %I1:L	0<L<17	不可以
开关量输出位	%Q0:L 或 %Q1:L	0<L<17	可以
系统位	%Si:L i 为 8 的倍数	0<L<17 和 i+L-128	根据 i
内部位	%Mi:L i 为 8 的倍数	0<L<17 和 i+L-128	可以

位串可被用于赋值指令 := （见赋值指令，节 3.1-4）。

字表

字表是指一系列类型相同且相邻的字，被定义长度为：L。

字表举例：

%KW10:7

	16 bits
%KW10	
%KW16	

类型	地址	最大范围	写访问
内部字	%MWi:L	0<L<512 和 i+L-512	可以
常量字	%KWi:L	0<L 和 i+L-64	不可以
系统字	%SWi:L	0<L 和 i+L-128	根据 i

字表可被用于赋值指令 := （见赋值指令，节 3.1-4）。

(1) %M8:6 可以，因为 8 是 8 的倍数；%M10:16 不可以，因为 10 不是 8 的倍数。

索引字

- 直接寻址

如果一个对象的地址是在编写程序时定义的并且固定不变, 那么该对象的地址被称为直接地址。
例如：%M26 （地址为 26 的内部位）

- 间接寻址

在间接寻址时， 在一个对象的直接地址中加入索引：把索引值加入对象地址。索引由内部字 %MWi 定义。“ 索引字 ” 的数量不限。

例如：%MW108[%MW2] 的地址为字的直接地址 108 + 字 %MW2 的值。
如果字 %MW2 的值为 12，则 %MW108[%MW2] 等效与 %MW120。

类型	地址	最大范围	写访问
内部字	%MWi[%MWj]	0-i+%MWj<512	可以
常量字	%KWi[%MWj]	0-i+%MWj<64	不可以

索引字可用于赋值指令 := （见赋值指令，节 3.1-4），也可用于比较指令（见比较指令，节 3.1-5）。
这种寻址方式允许在程序中调整索引字的值, 从而能够连续扫描相同类型的一系列对象(内部字, 常量字等)。

- 索引溢出，系统位 **%S20**

如果一个索引对象的地址超出了同类型对象内存区的范围, 就称为索引溢出。以下情况会发生索引溢出:

- 对象地址 + 索引值 小于 0。
- 对象地址 + 索引值 大于 512（对字 %MWi 而言）或者 63（对字 %KWi 而言）。
  如果发生索引溢出，系统将系统位 %S20 置为 1，对象将索引值置为 0。

注意：用户应该监视任何溢出：在处理过程中用户程序应读取位 %S20，以保证它被复位为 0。
%S20（初始状态为 0）:

- 发生索引溢出：由系统置为 1。
- 确认索引溢出：在修改了索引之后，由用户置为 0。

3.1-3 数字指令说明

数字指令一般用于 16 位字（见节 3.1-1），并写在方括号内。若前一个逻辑运算值为真（布尔累加器 = 1），则执行数字指令。若前一个逻辑运算值为假（布尔累加器 = 0），则不执行数字指令且操作数保持不变。

3.1-4 赋值指令

用于把操作数 Op2 装入操作数 Op1。

语法： [Op1:=Op2] <=> Op2->Op1

可对下列对象执行赋值操作：

- 位串
- 字
- 字表

位串赋值（见节 3.1-2 中的位串对象）

可用于位串的赋值操作：

位串 - > 位串（例 1）

位串 - > 字（例 2）

字 - > 位串（例 3）

立即值 - > 位串

例如

%Q0:8:=%M64:12

%I0.2 %MW100:=%I0:16

%I0.3 P %M104:16:=%KW0

```
LD      1
[%Q0:8:=%M64:12]          (例 1)

LD      %I0.2
[%MW100:= %I0:16]         (例 2)

LD      %I0.3
[%M104:16:= %KW0]         (例 2)
```

使用规则

- 对于位串 - > 字赋值：位串中的位从右开始传送到字（位串的第一位送到字的第 0 位），并把字中没有被传送的位都置为 0（长度 <16）。
- 对于字 - > 位串赋值：字中的位从右起开始传送（字的第 0 位关联位串的第一位）。

语法

操作	语法	操作数 1 (Op1)	操作数 2 (Op2)
:=	[Op1: = Op2] 把操作数 2 的值赋给操作数 1	%MWi,%QWi, %SWi %MWi[MWi], **%Mi:L,%Qi:L,%Si:L**	立即值，%MWi, %KWi,%IW,%QW,%SWi, %BLK.x,%MWi[MWi], KWi[MWi], **%Mi:L,%Qi:L,** **%Si:L,%Ii:L,**

注意：缩写 %BLK.x （例如 %C0.P ）可用来表示任意功能块字。

字赋值

可用于字的赋值操作：

字 -> 字 (例 1)

字 -> 索引字

索引字 -> 字

索引字 -> 索引字 （例 2)

立即值 -> 字 (例 3)

立即值 -> 索引字

位串 -> 字

字 -> 位串

例如：




```
LD      1
[%SW112:= %MW100]        (例 1)

LD      %I0.2
[%MW0[%MW10] := %KW0[%MW20]]
                         (例 2)

LD      %I0.3
[%MW10:= 100]            (例 3)
```

语法

操作	语法	操作数 1 (Op1)	操作数 2 (Op2)
:=	[Op1: = Op2] 把操作数 2 的值赋给操作数 1	**%MWi,%QWi, %SWi** **%MWi[MWi],** %Mi:L,%Qi:L,%Si:L,	立即数，**%MWi,** **%KWi,%IW,%QW,%SWi,** **%BLK.x,%MWi[MWi],** **%KWi[MWi],** %Mi:L,%QiL, %Si:L,%Ii:L,

注意：

- 缩写 %BLK.x （例如 R3.I）可用来表示任意功能块字。
- 对于位串 %Mi:L, %Si:L 和位串的第一个基地址应是 8 的倍数 (0, 8, 16, ..., 96, ...)。

字表赋值（见节 **3.1-2** 中的字表对象）

可用于字表的赋值操作：

立即值 - > 字表（例 1）

字 - > 字表（例 2）

字表 - > 字表（例 3）

例如


```
LD      1
[%MW0:10:= 100]          (例 1)

LD      %I0.2
[%MW0:10:=%MW11]         (例 2)

LD      %I0.3
[%MW10:20= %KW30:20]     (例 3)
```

语法

操作	语法	操作数 1 (Op1)	操作数 2 (Op2)
:=	[Op1: = Op2] 把操作数 2 的值赋给操作数 1	**%MWi:L,%SWi:L**	**%MWi,%KWi:L** **%SWi:L** 立即数,%MWi,%KWi,%IW,%QW,%SWi,%BLK.x

3.1-5 比较指令

比较指令用于比较两个操作数。

> : 测试操作数 1 是否大于操作数 2。

>= : 测试数 1 是否大于或等于操作数 2。

< : 测试操作数 1 是否小于操作数 2。

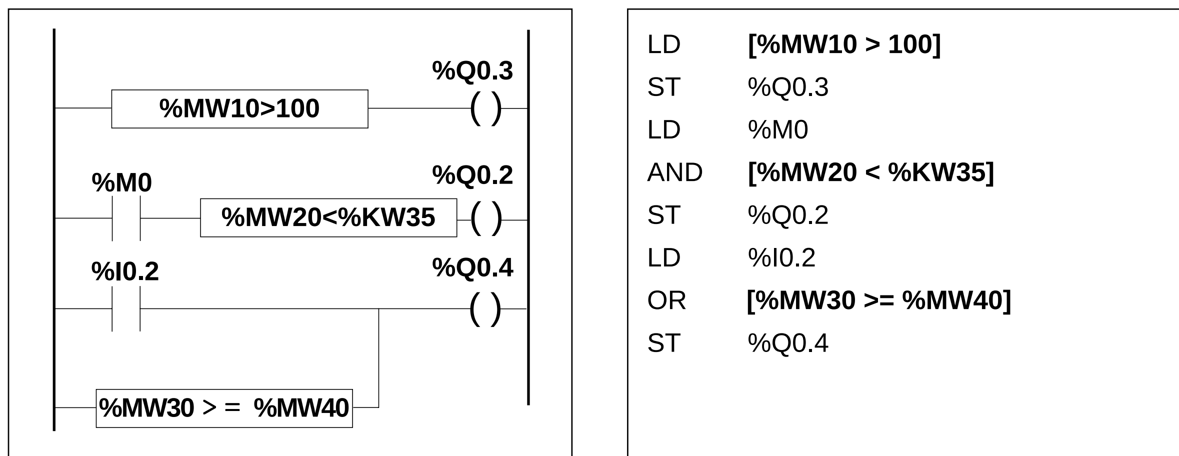
<= : 测试操作数 1 是否小于或等于操作数 2。

= : 测试操作数 1 是否等于操作数 2。

<> : 测试操作数 1 是否不等于操作数 2。

结构

比较指令要加方括号，跟在指令 LD、AND 和 OR 的后面。当比较结果为真时，值为 1。



语法

操作	语法	操作数 1 (Op1)	操作数 2 (Op2)
>,>=,<,<= =,<>	LD [Op1 运算符 Op2] AND[Op1 运算符 Op2] OR[Op1 运算符 Op2]	%MWi:L,%KWi,%IW, %QWi,%SWi,%BLK.x	立即数，%MWi,%KWi, %IW,%QW,%SWi,%BLK.x, %MWi[%MWi], %KWi[%MWi]

注意：

比较指令可以放在圆括号内。

例如：

```
LD      %M0
AND(    [%MW20>10]
OR      %I0.0
)
ST      %Q0.1
```

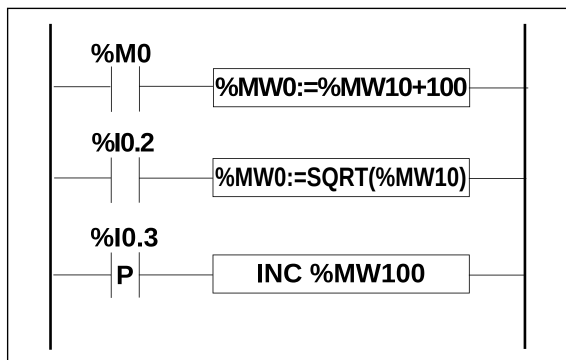
3.1-6 算术指令

这类指令用于两个操作数之间或一个操作数上的算术运算。

+	: 两个操作数相加	**REM**	: 两个操作数相除的余数
-	: 两个操作数相减	**SQRT**	: 一个操作数的平方根
*	: 两个操作数相乘	**INC**	: 一个操作数递增
/	: 两个操作数相除	**DEC**	: 一个操作数递减

结构

算术操作如下表示：




```
LD      %MW0
[%MW0:= %MW10+100]
LD      %I0.2
[%MW0:= SQRT(%MW10)]
LDR     %I0.3
[INC  %MW100]
```

语法：

取决于使用的运算符：见下表。

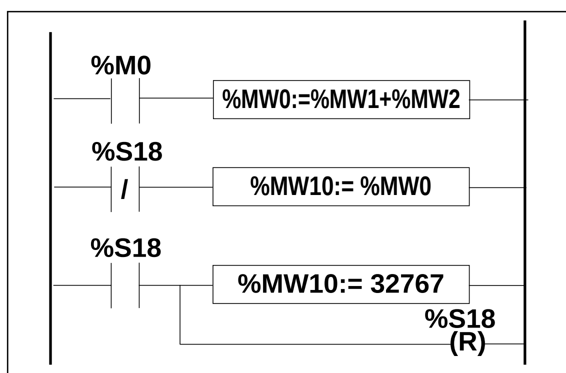
操作	语法	操作数 1 (Op1)	操作数 2 (Op2)
+,-,*,/,REM	[Op1: =Op2 运算符 Op3]	%MWi,%QWi,%SWi,	立即数(1),%MWi,%KWi %IW,%QW,%SWi,%BLK.x
SQRT	[Op1: =SQRT(Op2)]		
INC,DEC	[运算符 Op1]		

使用规则

- 加法：运算时溢出

如果结果超出 -32768 或 +32767，位 %S18（溢出位）置为 1，且所得结果不正确（见下页）。可由用户程序控制 %S18。

例如：




```
LD      %M0
[%MW0:= %MW1+%MW2]
LDN     %S18
[%MW10:= %MW0]
LD      %S18
[%MW10 := 32767]
R       %S18
```

如果 %MW1 =23241 且 %MW2=21853，实际结果 (45094) 不能由 16 位字显示， 位 %S18 被置为 1， 并且得出错误结果 (-20442)。在该例中，当结果值大于 32767 时，都取为 32767。

(1) 在 SQRT 中，操作数不能是立即值。

结果绝对溢出（无符号运算）：
在某些运算中可能会用到无符号操作数（此时第 15 位表示值 32768）。无符号操作数的最大值为 65535。若两个绝对值相加的和超过 65535，那么结果溢出。该情况由位 %S17 作为标记。当结果为 65536 时，%S17 变为 1。

例 **1**: [%MW2:=%MW0 + %MW1] 其中 %MW0 =65086, %MW1=65333
字 %MW2 的值为 64883。位 %S17 置为 1，表示值 65536。无符号运算结果等于：
65536 + 64883 = 130419。
例**2**: [%MW2:=%MW0 + %MW1] 其中 %MW0 =45736（等于带符号值 -19800）, %MW1=38336（等于带符号值 27200）。
系统位 %S17 和 %S18 被置为 1。带符号运算的结果 (+18536) 是错误的。无符号运算的结果（18536 + %S17 的值，等于 84072）是正确的。

- 减法：

负值
如果减法的结果小于 0，系统位 %S17 置为 1。

- 乘法：

运算时溢出
如果结果超出字的范围，位 %S18（溢出）置为 1，且结果无意义。

- 除法／取余：

被 0 除
如果除数为 0，则不能运算，而且系统位 %S18 置为 1，结果出错。
运算时溢出
如果商超出字范围，位 %S18 置为 1。

- 平方根开方：

只有正数才能进行平方根开方，所以通常结果为正。如果操作数为负，系统字 %S18 置为 1，结果出错。

注意：
用户程序可以控制系统字 %S17 和 %S18。如果它们被 PLC 置为 1，则程序应将它们复位，以便再次使用（见上页举例）。

3.1-7 逻辑指令

这些指令用于两个操作数之间或对于一个操作数进行逻辑运算。

AND: 与　　　用于两个操作数之间

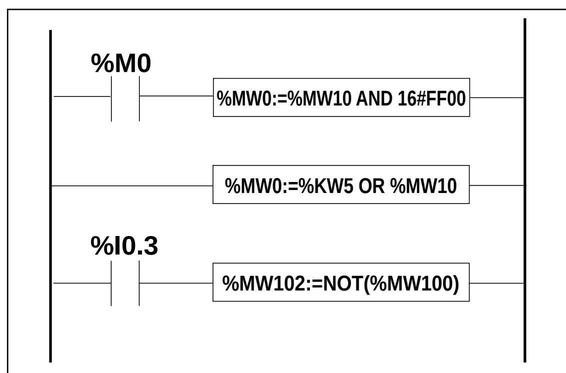
OR ：逻辑或　用于两个操作数之间

XOR: 异或　　用于两个操作数之间

NOT: 逻辑反　用于一个操作数

结构

逻辑运算表示如下：




```
LD      %M0
[%MW0:= %MW10 AND 16#FF00]
LD      1
[%MW0:= %KW5 OR %MW10]
LD      %I0.3
[%MW102:=NOT (%MW100)]
```

语法

取决于使用的运算符：见下表。

操作	语法	操作数 1 (Op1)	操作数 **2 (Op2 & Op3)**
AND,OR,XOR	[Op1: =Op2 运算符 Op3]	%MWi,%QWi,%SWi,	立即数(1)，%MWi,%KWi
NOT	[NOT (Op2)]		%IW,%QW,%SWi,%BLK.x

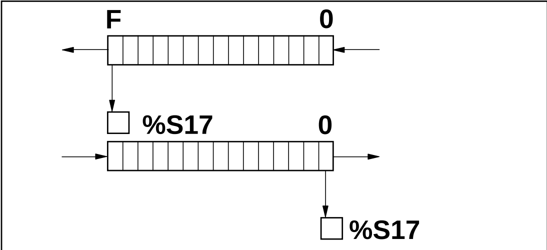
例如：[%MW15:=%MW32 AND %MW12]

(1) 在 NOT 指令中, Op2 不可以是立即值。

3.1-8 移位指令

移位指令可以把操作数的位向左或向右移动若干位。

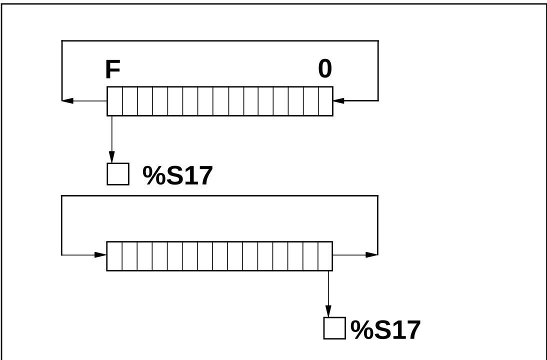
有两类操作指令:



• 逻辑移位:

- **SHL**(op2,**i**) 向左逻辑移动 i 位

- **SHR**(op2,**i**) 向右逻辑移动 i 位



• 循环移位:

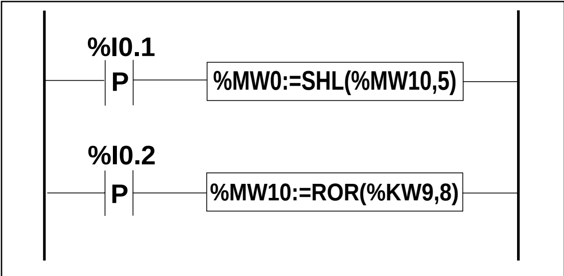
- **ROL**(op2,**i**) 向左循环移动 i 位

- **ROR**(op2,**i**) 向右循环移动 i 位

如果操作数时单字长的，则 i 的范围是 1 到 16。
最后输出位的状态被存在位 %S17 中。

结构

逻辑运算表示如下:




```
LDR     %I0.1
[%MW0 := SHL(%MW10,5)]

LDR     %I0.2
[%MW10 :=ROR (%KW9,8)]
```

语法

取决于使用的运算符：见下表。

操作	语法	操作数 1 (Op1)	操作数 **2 (Op2)**
SHL,SHR, ROL,ROR	[Op1: = 运算符(Op2,i)]	%MWi,%QWi,%SWi,	%MWi,%KWi, **%IW,%QW,%SWi,%BLK.x**

3.1-9 转换指令

有两种类型的转换指令：

- **BTI:** BCD--＞ 二进制 转换
- **ITB:** 二进制 --> BCD 转换

BCD 码介绍

BCD 码用四位二进制码表示一个十进制数（0 到 9)。一个 16 位字可用 4 个十进制数表示（0 到 9999)。

在转换中，如果值不是 BCD 码，则位 %S18 置为 1。可在用户程序中测试该位并将其复位为 0。

十进制数	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

例如：

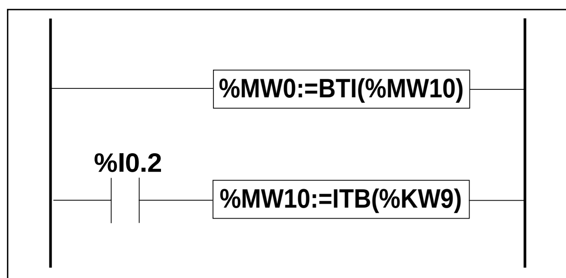
- 字 %MW5 用 BCD 的表示为“ 2450 ”，相应的二进制码为： 0010 0100 0101 0000
- 字 %MW12 用十进制表示为：“ 2450 ”，相应的二进制码为： 0000 1001 1001 0010

用指令 BTI 把字 %MW5 转换成 %MW12。

用指令 ITB 把字 %MW12 转换成 %MW5。

结构

转换运算表示如下：




```
LD      %M0
[%MW0 := BTI (%MW10)]

LD      %I0.2
[%MW10 :=ITB (%KW9)]
```

语法

取决于使用的运算符：见下表。

操作	语法	操作数 1 (Op1)	操作数 **2 (Op2)**
BIT,ITB,	[Op1: = 运算符（Op2)]	%MWi,%QWi,%SWi,	%MWi,%KWi,%IW,%QW, %SWi,%BLK.x

应用举例

BTI 指令用来处理旋转编码的 BCD 码在 PLC 输入的设定值。

ITB 指令用 BCD 码的数字值（例如，计算结果，功能块的当前值）。

3.2 专用功能块

3.2-1 与专用功能块相关的字和位对象

专用功能块使用和标准功能块类型相同的专用字和位（见节 2.2）。

程序可访问的功能块位和字对象

专用功能块		相关字和位	地址	写访问	参考节
脉冲宽度调制输出	字	占空比	%PWM.R	可以	3.2-3
%PWN		预设周期	%PWM.P	不可以	
脉冲发生器	字	预设值	%PLS.P	可以	3.2-4
%PLS		产生的脉冲数	%PLS.N	可以	
	位	当前输出	%PLS.Q	不可以	
		完成输出	%PLS.D	不可以	
高速计数器	字	阈值 i	%FC.Si	可以	3.2-5
%FC		当前值	%FC.V	不可以	
		预设值	%FC.P	可以	
	位	溢出	%FC.F	不可以	
		超时阈值 i 输出	%FC.THi	不可以	
发送报文	位	错误命令输出	%MSG.E	不可以	3.2-6
%MSG		命令完成输出	%MSG.D	不可以	
移位寄存器 %SBRi (i=0 到 7)	位	寄存器位 j=0~15	%SBRi.j	可以	3.2-7
步进计数器 %SCi (i=0 到 7)	位	步进计数器位 j=0~255	%SCi.j	不可以	3.2-8

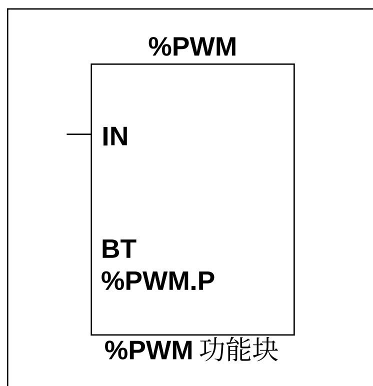
3.2-2 编程原则

与标准功能块相同，特殊功能块可有两种编程方法 :

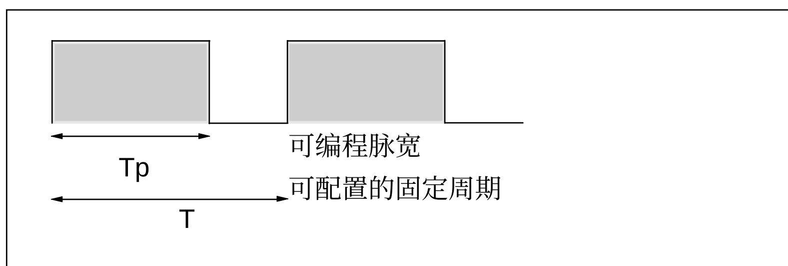
- 不可逆编程 : 使用特殊指令
- 可逆编程 : 仿照梯形图语言功能块

见节 2.2-2

3.2-3 脉冲宽度调制输出 % PWM



%PWM 功能块用于在 PLC 输出 %Q0.0 上生成一个方波信号。信号宽度（导通周期）可由程序改变。



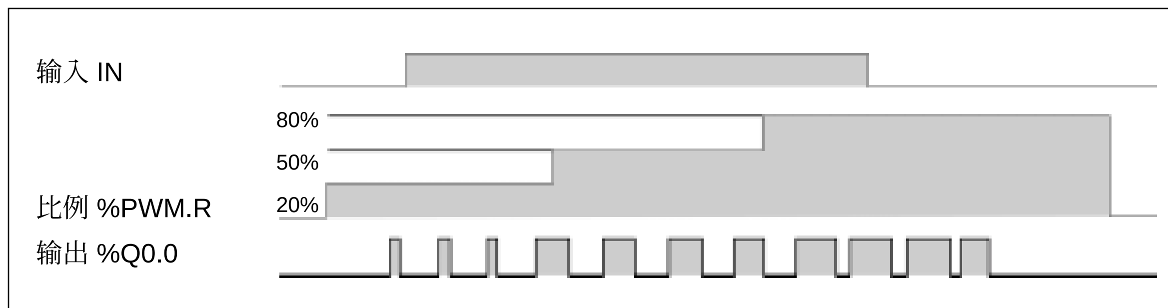
特性		
时基	**TB**	0.1 毫秒 (1), 10 毫秒, 1 秒（默认值）
预设周期	**%PWM.P**	0<%PWM.P-32767 如果时基为 10 毫秒或 1 秒 0<%PWM.P-255 如果时基为 0.1 毫秒 （0= 功能块未被使用） 预设值和时基可在配置时进行修改。它们用于确定信号周期 T = %PWM.P x TB。 频率越低，所选的 %PWM.P 值应越大。 所得周期的范围： • 时基为 0.1 毫秒时，0.2 到 26 毫秒 • 时基为 10 毫秒时，20 毫秒到 5.45 分 • 时基为 1 秒时，2 秒到 9.1 个小时
占空比	**%PWM.R**	0-%PWM.R-100 (2)，这个字表示状态 1 在信号周期中所占的比例（0 = 默认值）。 因此，脉宽 Tp 等于：Tp = Tx (%PWM.R/100) 字 %PWM.R 由用户程序写入，用于进行脉宽调制。
脉冲发生输入 （或指令）	**IN**	状态为 1 时，脉冲宽度调制信号由输出 %Q0.0。 状态为 0 时，输出 %Q0.0 置为 0。

(1) 该时基只适用于带晶体管输出的 TSX Neza PLC。

(2) 如果值大于 100 则看作等于 100。

操作

输出信号 %Q0.0 的频率在配置时通过选择时基 TB 和预设 %PWM.P 来设置。信号的带宽通过改变程序中的 %PWM.R 比例来调节。



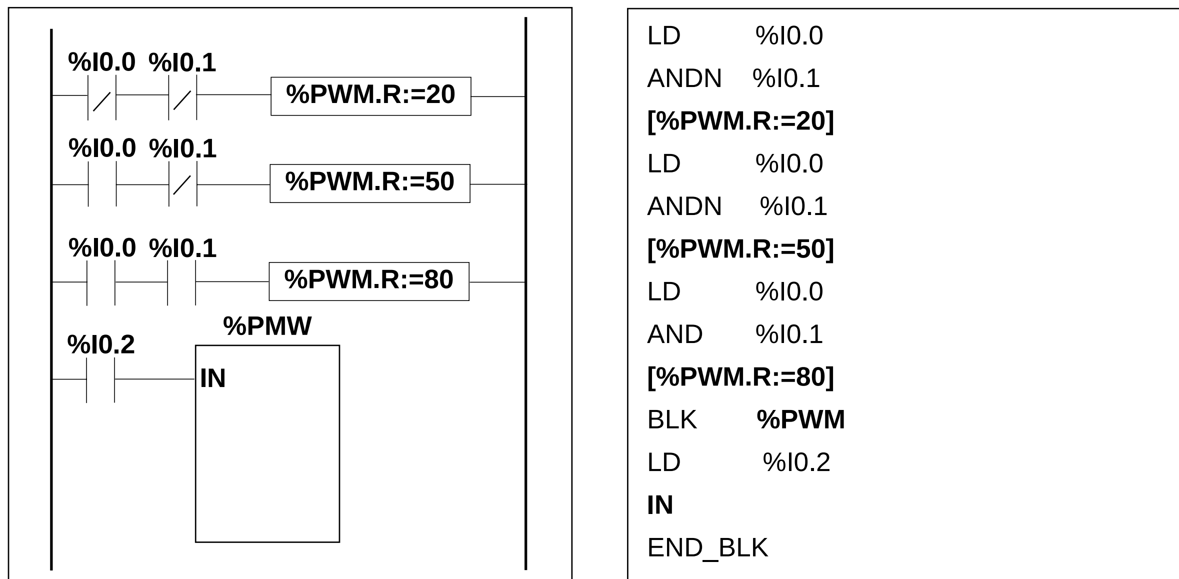
编程和配置

在这个例子中，信号的带宽由程序根据 PLC 输入 %I0.0 和 %I0.1 的状态进行修改。在配置时把信号的周期设为 500 毫秒。（见下面配置记录）

如果 %I0.0 和 %I0.1 都设为 0，%PWM.R 比例设为 20%，则状态 1 的持续时间为 : 20% x 500 毫秒 = 100 毫秒。

如果 %I0.0 设为 0，%I0.1 设为 1，%PWM.R 比例设为 50%，则持续时间为 250 毫秒。

如果 %I0.0 和 %I0.1 都设为 1，%PWM.R 比例设为 80%，则持续时间为 400 毫秒。



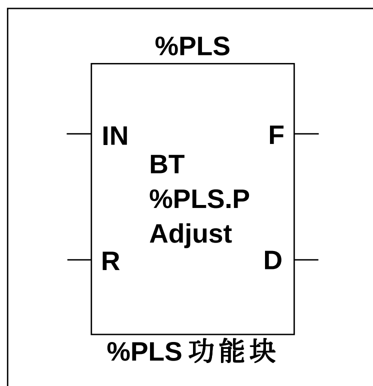
配置

输出 %Q0.0 = 输出 %PWM TB = 10 毫秒 %PWM.P=50

特殊情况

- 冷启动的影响 :(%S0=1) 设 %PWM.R 比例为 0
- 热启动的影响 :(%S1=1) 无影响
- **PLC** 停止的影响：
  不论系统位 %S8 的状态如何，输出 %Q0.0 设为 0。
- 时基为 **0.1** 毫秒，通过编程设备强置输出 %Q0.0 不会中止信号的生成。

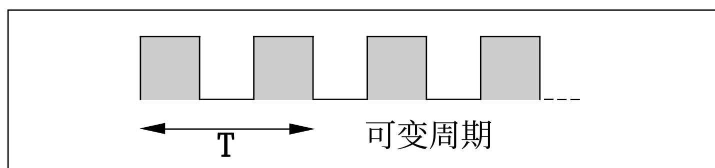
3.2-4 脉冲发生器输出 % PLS



%PLS 功能块在 PLC 输出 %Q0.0 上生成一个方波信号。(50% 的占空比)

该信号可以：

- 限制长度，由程序（或配置时）确定脉冲的个数及周期。
- 不限制长度，由程序（或配置时）确定脉冲的周期。

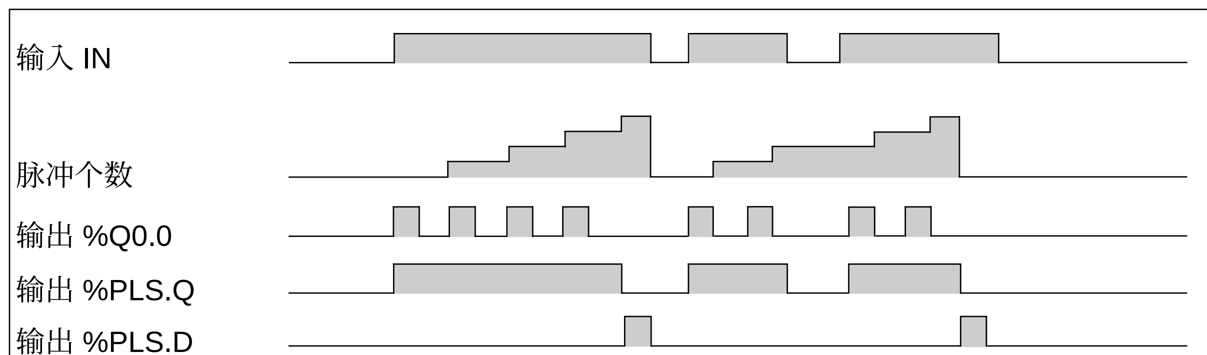


特性		
时基	**TB**	0.1 毫秒 (1)，10 毫秒，1 秒（默认值）
预设值	**%PLS.P**	0<%PLS.P<32767 如果时基为 10 毫秒或 1 秒 0<%PLS.P<255 如果时基为 0.1 毫秒 (1)(2) 预设值用于修改信号周期 T = %PWM.PxTB。 所得周期的范围： • 时基为 0.1 毫秒时，0.2 到 26 毫秒 • 时基为 10 毫秒时，20 毫秒到 5.45 分钟 • 时基为 1 秒时，2 秒到 9.1 个小时 注意：**%PLS.P** 必须是个偶数
脉冲个数	**%PLS.N**	0<%PLS.N<32767，这个字表示要生成的脉冲列中脉冲的个数。0 = 无限长方波信号（默认值）。 %PLS.N 由程序测试和写入。
根据编程终端调整	**Y/N**	Y：可以在数据编辑器中修改预设值 %PLS.P.N N：不可访问数据编辑器，无法修改
脉冲发生输入（或指令）	**IN**	状态为 1 时，输出 %Q0.0 处生成信号。 状态为 0 时，输出 %Q0.0 置为 0。
复位输入（或指令）	**R**	状态为 1 时, 把输出 %PLS.Q 和 %PLS.D 的脉冲数复位为 0。
当前脉冲生成输出	**%PLS.Q**	状态为 1，在 %Q0.0 处生成脉冲信号。
脉冲生成完成输出	**%PLS.D**	状态为 1，脉冲输出已经完成。
PLS 计数器 (2)		N= 否, Y= 是 该选项允许输入 %I0.0 被用作计数输入。

(1) 该时基只适用于带晶体管输出的 TSX Neza PLC。

(2) 当时基为 0.1 毫秒时需选择 PLS 计数器选项。为了使 PLS 正常工作, 必须在物理上把输出 %Q0.0 和输入 %I0.0 连接起来。在这类操作中，%PLS.P 必须大于或等于 6（最大频率为 1.6 KHz），以保证功能块的正常运行。

操作



特殊情况

· 冷启动的影响 :(%S0=1) 把 %PLS.P 的值设为配置时定义的值
· 热启动的影响 :(%S1=1) 无影响
· **PLC** 停止的影响：见节 3.2-3 中的 PLC 停止的影响
· 改变 **%PLS.P** 预设值的影响：通过指令修改 %PLS.P 的预设值或对它进行调整会立即产生影响。
· 时基为 0.1 毫秒，强制输入 %QI0.0 不会中止信号的生成。

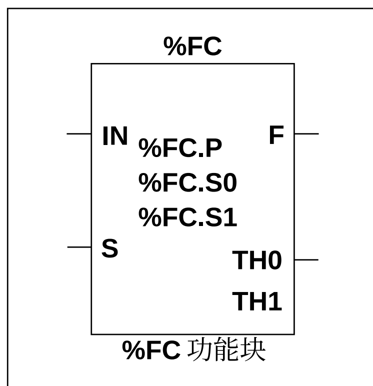
注释：

使用 0.1 毫秒的时基可能造成要求脉冲个数 %PLS.N 与实际脉冲个数相差一个脉冲。为了避免这种情况，当一次脉冲生成完成后必须复位功能块。

```
LD    %Mi
IN    %PLS
N
R     %PLS
```

然而，一旦输入 IN 变为 1，那么将会重新开始生成 %PLS.N 个脉冲。

3.2-5 高速计数功能块，频率计和加／减计数器 %FC



%FC 功能块

%FC 功能块可用于执行以下三个功能：
高速计数
频率计
加／减计数
（详见 A 部分，节 4.4）在配置时选择功能。
注意：可不通过编程而是直接配置 I/O 和参数来执行高速计数功能。

功能块 **%FC** 的特性

该功能块提供一组字和一些用于生成三种功能的输入输出位。要充分了解每个对象在功能块中的作用，请参考功能块的详细介绍。

当前值	**%FC.V**	根据所选输入及功能增加或减少 可被读取和测试，但不可写入
预设值	**%FC.P**	只供加／减计数器功能使用 0-%FC.P-65535 字可被读取、测试和写入
阈值 S0 (1)	**%FC.S0**	0-%FC.S0-65535 字表示配置时阈值 0 的值 可由程序读取和写入
阈值 S1 (1)	**%FC.S1**	0-%FC.S1-65535 字表示配置时阈值 1 的值 可由程序读取和写入
输入使能 （或指令）	**IN**	状态为 1，激活当前功能 状态为 0，停止当前功能
设置输入 （或指令）	**S**	状态为 1： • 把当前值设为预设值（加／减计数器）或将当前值复位为 0。 • 初始化阈值输出操作 (HSC_Out) • 考虑用程序修改阈值 %FC.S0 和 %FC.S1
溢出	**%FC.F**	若 %FC.V 的值超过 65535，则状态为 1。 可用预设值刷新（%I0.1 或 S 指令）或者冷启动。
阈值 位 0 (2)	**%FC.TH0**	若当前值大于或等于阈值 %FC.S0,则状态为 1。
阈值 位 1 (2)	**%FC.TH1**	若当前值大于位 1 (2) 或等于阈值 %FC.S1，则状态为 1。

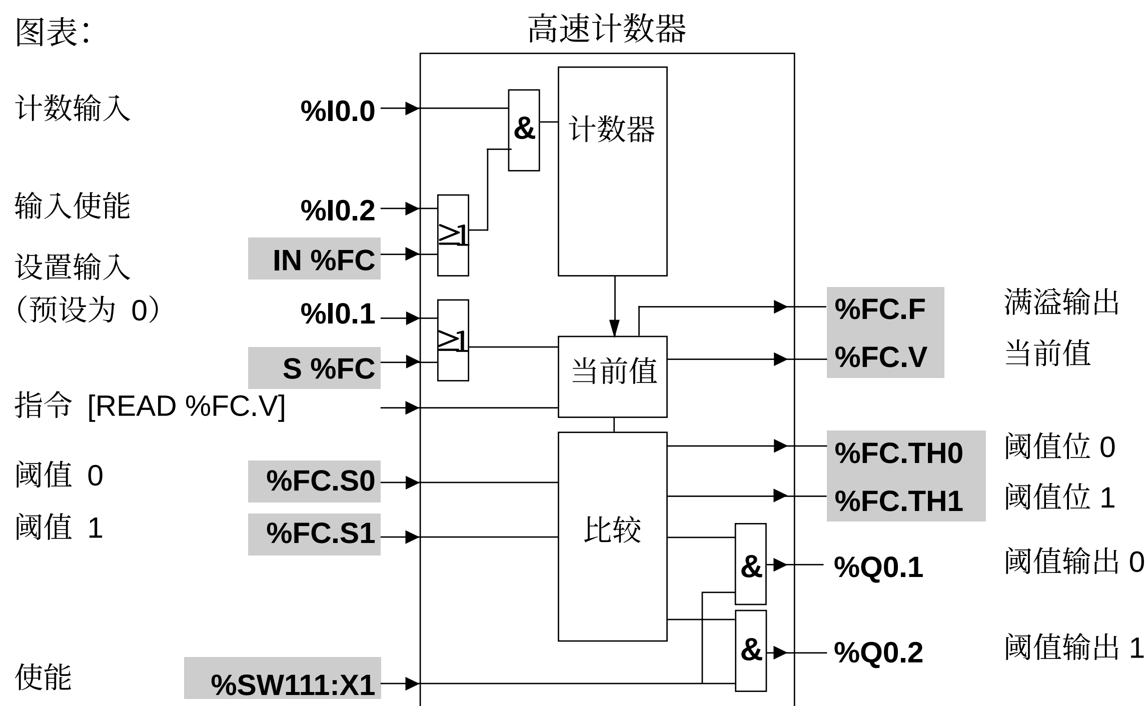
(1) 阈值 %FC.S0 应小于阈值 %FC.S1。
(2) 程序中只能测试一次 %FC.THx 位。

高速计数功能

高速计数器在高速模式下最高频率可达 10 kHz （正常模式下为 5 kHz，可在配置时选择），范围为 1- 65535。

计数器对 PLC 输入 %I0.0 的信号进行计数。计数值（当前值 %FC.V）和预设的两个阈值 %FC.S0 和 %FC.S1 作比较，这两个阈值在配置时定义并可由程序修改（修改时设置输入被激活）。

图表：



注释：除了计数输入 %I0.0 外，所有其它功能块的预设 I/O 都可选（可在配置时选择）。它们每个都有等效的编程对象和指令（上图中由方框标出）。

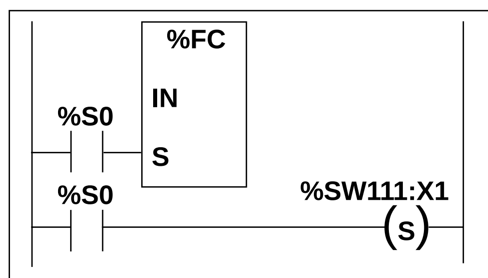
阈值输出：阈值输出直接由高速计数器根据用户配置时所设值来控制（不用等到扫描结束时更新输出）。

输出	FC.V<FC.S0<FC.S1	FC.S0-FC.V-FC.S1	FC.S0<FC.S1<FC.V
%Q0.1	0 或 1	0 或 1	0 或 1
%Q0.2	0 或 1	0 或 1	0 或 1

阈值输出初始化后，必须由高速计数器的预设命令来启动。

编程举例：

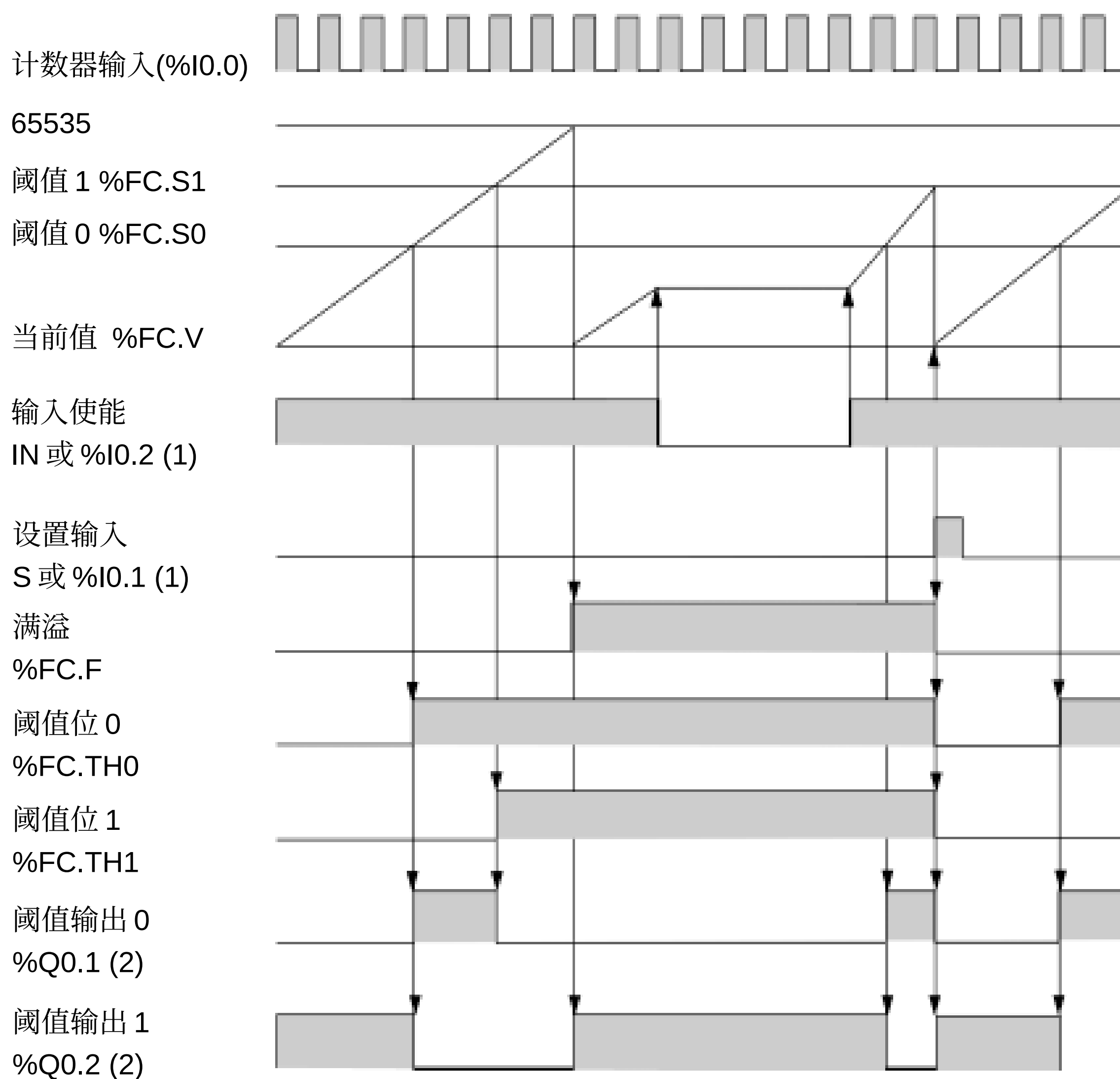
```
BLK  %FC
LD  %S0
S
END_BLK
LD  %S0
S  %SW111:X1 (把 %SW111:X1 设置为 1)
```



读取当前值

当前值 %FC.V 在每次 PLC 扫描结束处更新。%FC.V 也可由指令 READ 更新：

语法 :[READ %FC.V] 时序图



(1)当设置输入 S 动作时，输入 %I0.1 上升沿有效（见时序图）

(2)在此时序图中，状态如下表所示：

输出	FC.V<FC.S0<FC.S1	FC.S0-FC.V-FC.S1	FC.S0<FC.S1<FC.V
%Q0.1	0	1	0
%Q0.2	1	0	0

- 频率计功能

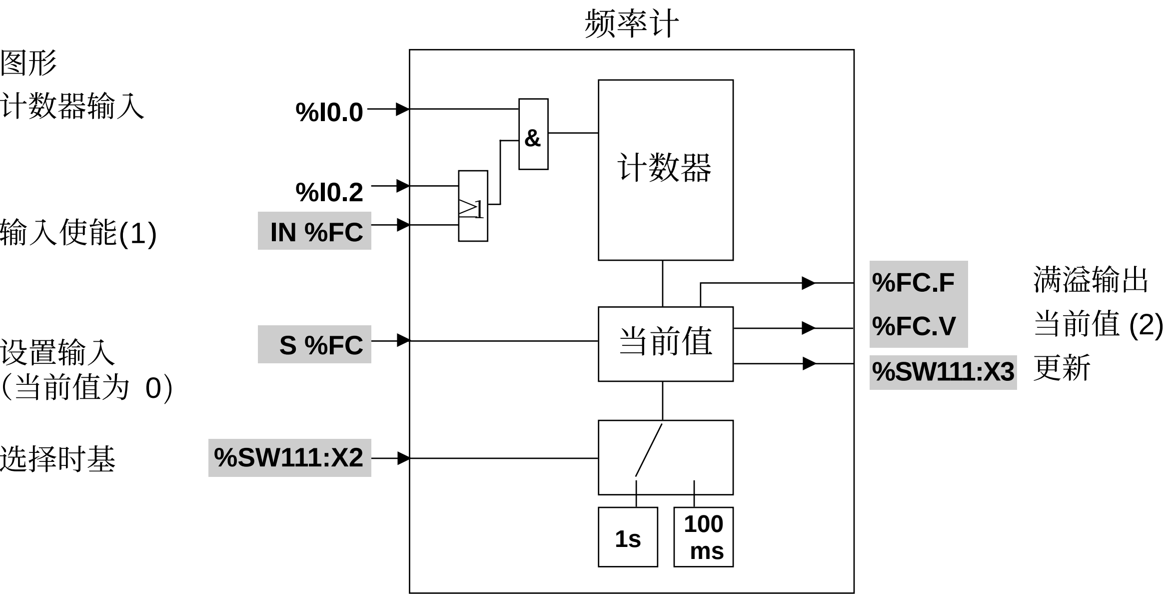
频率计用来测试输入 %I0.0 上的单位为 Hz 的周期信号频率。有两种模式：高速模式(10kHz 滤波)和正常模式(5kHz 滤波)。

频率测试范围从 1Hz 到 10kHz。用户可通过系统位 %SW111:X2 在两个时基中选择一个。

(1 表示 100 毫秒的时基，0 表示 1 秒的时基)。

时基	测试范围	测试范围	更新
100 毫秒	100Hz-10kHz	10kHz 时 0.1% 100Hz 时 10%	每秒 10 次
1 秒	10Hz-10kHz	10kHz 时 0.01% 10Hz 时 10%	每秒 1 次

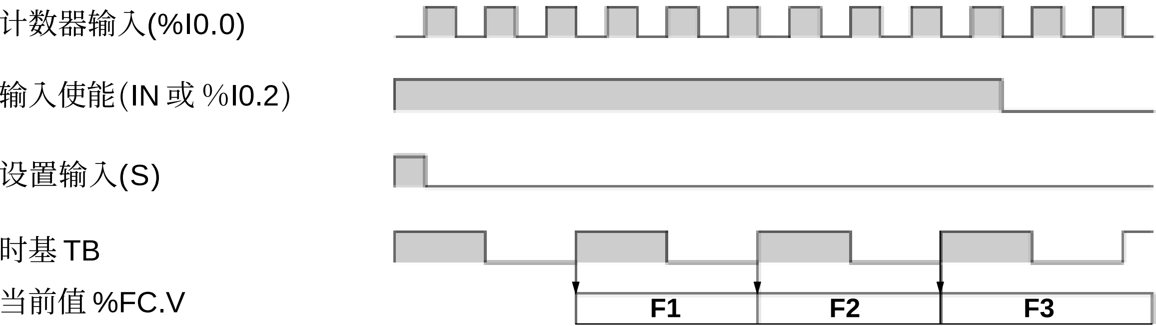
每次当前值更新系统位 %SW111:X3 都置 1。用户程序必须将系统位复位为 0。



(1) 输入 %I0.2 可选，可在配置时选择。

(2) 当前值 FC.V 的单位为 Hz。

时序图



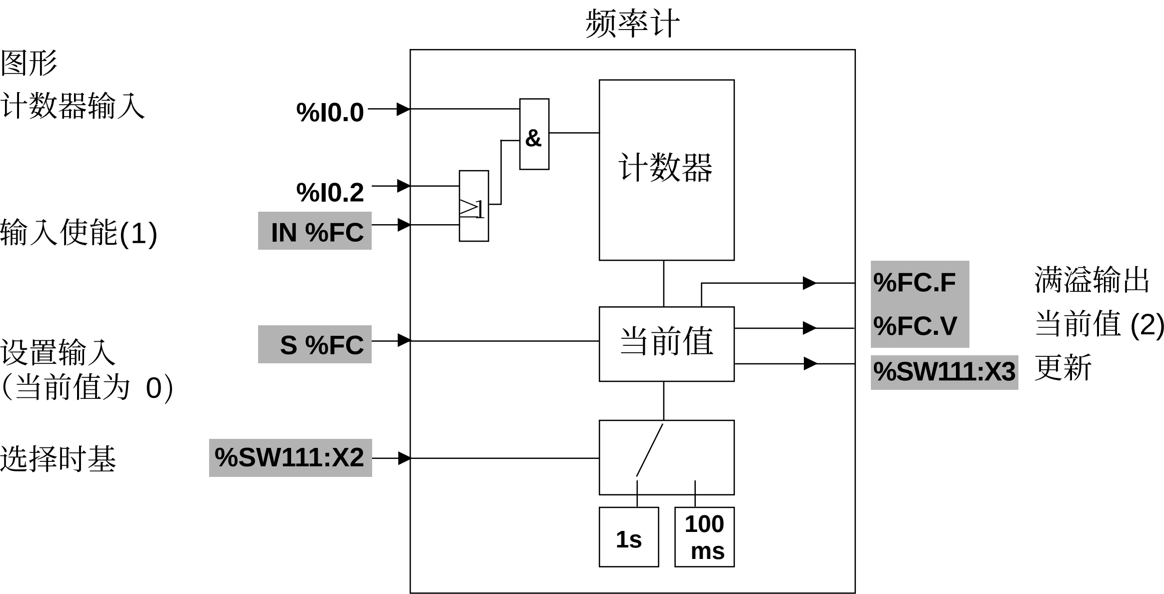
- 频率计功能

频率计用来测试输入 %I0.0 上的单位为 Hz 的周期信号频率。有两种模式：高速模式(10kHz 滤波)和正常模式(5kHz 滤波)。

频率测试范围从 1Hz 到 10kHz。用户可通过系统位 %SW111:X2 在两个时基中选择一个。(1 表示 100 毫秒的时基，0 表示 1 秒的时基)。

时基	测试范围	测试范围	更新
100 毫秒	100Hz-10kHz	10kHz 时 0.1% 100Hz 时 10%	每秒 10 次
1 秒	10Hz-10kHz	10kHz 时 0.01% 10Hz 时 10%	每秒 1 次

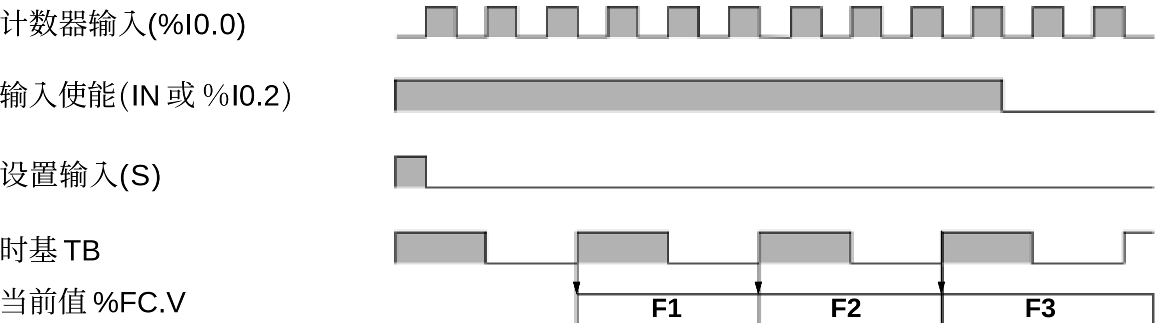
每次当前值更新系统位 %SW111:X3 都置 1。用户程序必须将系统位复位为 0。

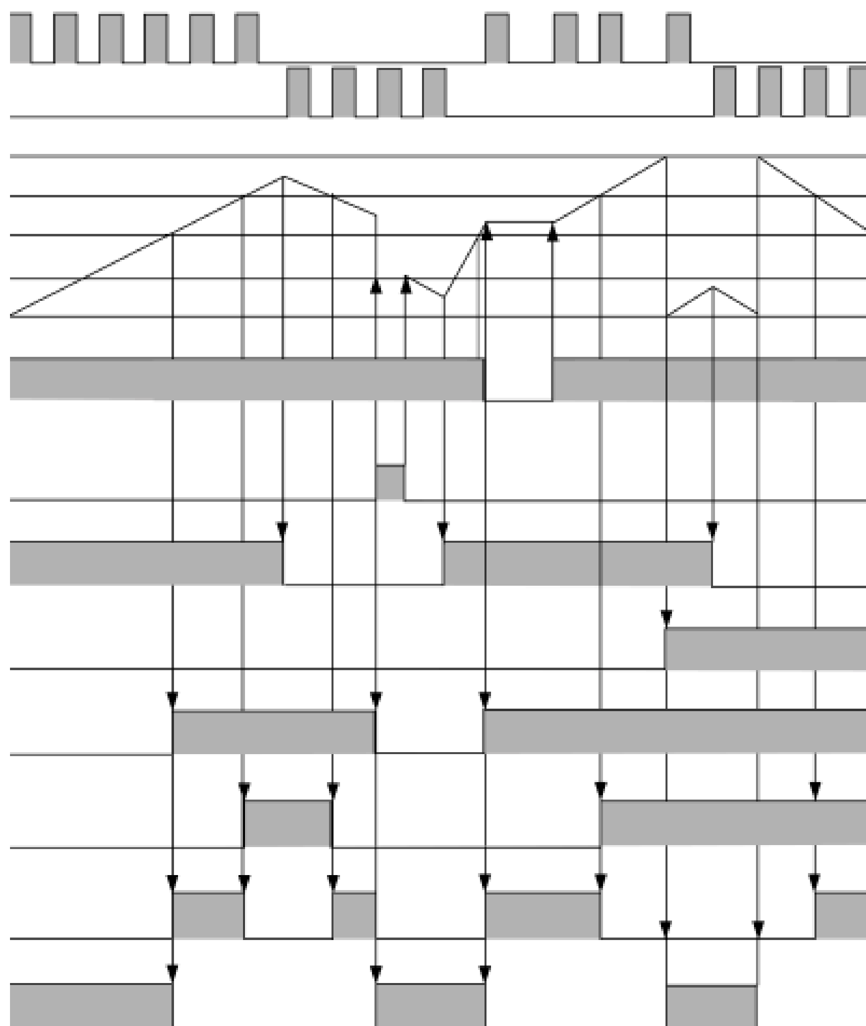


(1) 输入 %I0.2 可选，可在配置时选择。

(2) 当前值 FC.V 的单位为 Hz。

时序图





(1) 输入 %I0.1 为上升沿。

(2) 计数器初始化后，满溢输出 %FC.F 复位为 0。

(3) 参考高速计数器的状态矩阵。

要点：

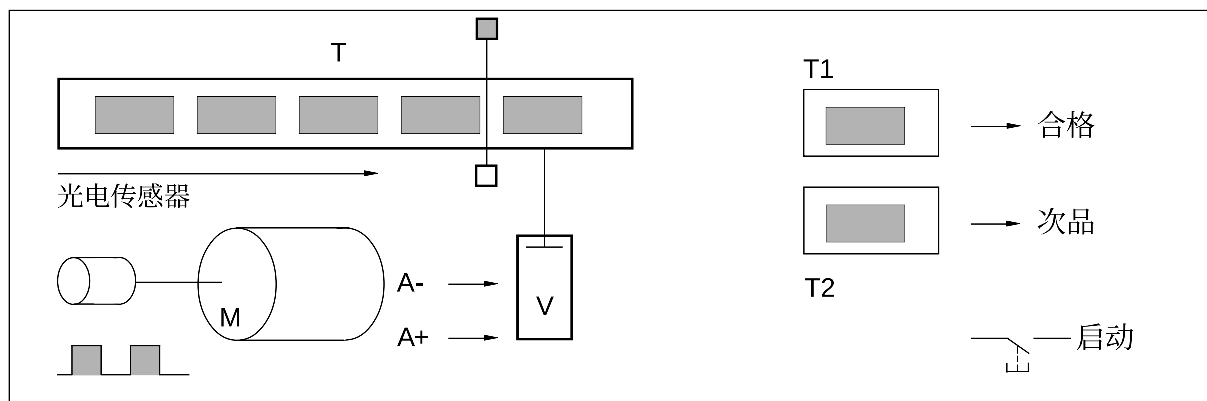
> 如果位于上升或下降沿的 4 个输入 %I0.0、%I0.1、%I0.3 和 %I0.4 中的 1 个状态为 1， 则其他 3 个输入不执行。

特殊情况（高速计数器，频率计和加／减计数器）

- 冷启动的影响 :(%S0=1) 改变当前值，改变输出 %FC.F，%FC.TH0，%FC.TH1，把阈值输出 (%SW111:X1) 位设为 0，并把配置值复制到字 %FC.P 和 %FC.S0/S1。
- 热启动的影响 :(%S1=1) 和 PLC 中止：对当前值无影响。

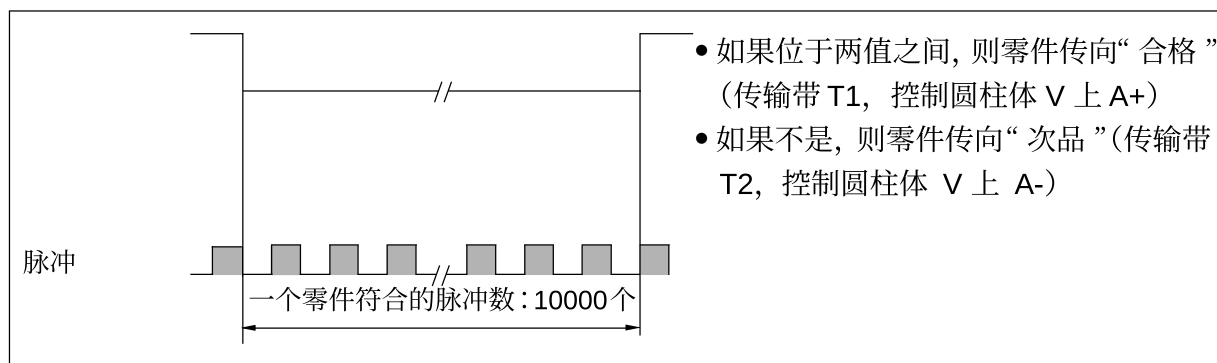
高速计数功能举例

应用介绍



零件被放在传输带上依次进行测试，传输带由马达带动，而马达和一个增量型旋转编码器连接。通过对移动零件在单位时间内产生的脉冲数计算来进行测试。圆柱体 V 控制皮带 T 的运动，根据测试结果决定 T 应位于 T1 还是 T2。

测试到的脉冲数与两个极限值比较（长度测量公差）。



(“ 启动 ” 按钮用于启动整个操作。)

I/O 赋值表

输入

- %I0.0 计数器输入（连接增量型编码器）
- %I0.1 对应光电传感器的复位输入
- %I0.2 输入使能对应 “ 启动 ” 按钮

输出

- %Q0.1 圆柱体 A+ 控制输出
- %Q0.2 圆柱体 A- 控制输出
- %Q0.0 皮带控制输出

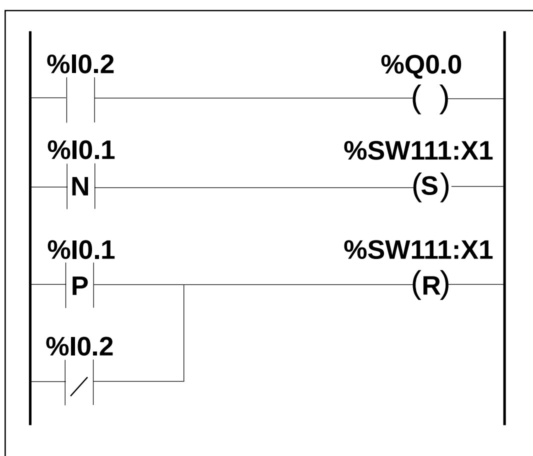
应用处理

高速计数功能可以无需 PLC 编程，直接由功能块 %FC 配置得到。

%FC :计数器
模式 :高速
计数输入 : %I0.0
设置 : %I0.1
输入使能 : %I0.2
%FC.S0 : 9950 阈值 0 对应最小容许值
%FC.S1 : 10000 阈值 1 对应最大容许值

输出矩阵

输出	FC.V<FC.S0<FC.S1	FC.S0<FC.V<FC.S1	FC.S0<FC.S1<FC.V
%Q0.1	0	1	0
%Q0.2	1	0	1




```
LD    %I0.2
ST    %Q0.0
LDF   %I0.1
S     %SW111:X1
LDR   %I0.1
ORN   %I0.2
R     %SW111:X1
```

操作

输出使能 %I0.2 (启动钮) 被激活，输入 %I0.0 立刻记录增量编码器生成的脉冲数。在输入 %I0.1 的上升沿，计数器当前值复位为 0。当光电传感器（输入 %I0.1)发现传输带上没有零件时，输出 %Q0.1 和 %Q0.2 被激活（通过位 %SW111:X1)，并根据计数器 %FC.V 的当前值（由输出矩阵而得）置为 0 或 1。

当零件在容许范围内，输出 %Q0.1 置为 1。由它控制传输带 T1 上圆柱体 A 的位置输出（见输出矩阵)。

当零件超出容许范围时（圆柱体 A 返回传输带 T2)，输出 %Q0.2 置为 1。

3.2-6 发送／接收报文和控制数据交换

TSX Neza 可以和一个编程终端进行通信（掌上电脑编程器或 PL707 WIN 软件）也可以和 TER 终端口上连接的其他 UNI-TELWAY 设备进行通信。TSX Neza 还可以在特性模式（ASCII 通讯方式）下配置为发送和／或接收报文。

PL707WIN 语言提供：

- EXCH 指令来发送／接收报文，
- %MSG 功能块来控制数据交换。

TSX Neza 根据电缆的旋转拨码开关来决定协议，并把该信息提供给系统位 %S100（/DPT 管理）。

关于 UNI-TELWAY 的兼容产品及配置可参见通讯部分。

EXCH 指令

SEND 指令允许 TSX Neza 从／向 UNI-TELWAY 或 ASCII 设备发送／接收信息。用户建立一个字表（%MWi:L 或 %KWi:L），其中包括要发送和／或接收的数据（最多可达 64 个数据字）。每种协议（ASCII 和 UNITELWAY）都有自己的字表格式。

可用 EXCH 指令进行信息交换。

语法 :**[EXCH %MWi:L] (1)** 或 **[EXCH %KWi:L]**

注意：某些支持 UNI-TELWAY 协议的设备如 XBT 或 CCX 人机界面，可以无需特别编程就可从 TSX Neza 发送或接收信息。TSX Neza 发送或接受信息。TSX Neza 必须在完成第一条 EXCH 指令后，才能继续执行第二条。当发送几条指令时，必须使用 %MSG 功能块。

(1) L:字表中的个数。

内部字表 %MWi:L 的值应如 i+L<=512。

%MSG 功能块

%MSG 功能块是可选的；它可以用来控制数据交换。%MSG 功能块有三个用途：

- 通信错误校验

错误校验用于校验 EXCH 指令确定的块长度（字表）是否足够装入要发送的信息（与字表中第 1 个字的低位有效字节的长度相比）。它也用于校验一条 UNI-TELWAY 信息是否被 UNI-TELWAY 设备接收到。

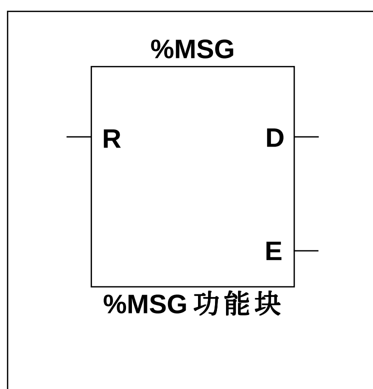
- 多条报文协调

在发送多条报文时 %MSG 功能块提供有关前一条报文是否完成的信息, 以保证多条报文发送时不发生冲突。

- 优先报文传送

%MSG 功能块允许暂停当前报文的传送，以立即发送紧急报文。

数据交换控制块



%MSG 功能块用于控制数据交换。
它在编程中可选。

%MSG 块的特性

输入复位 （或指令）	R	状态为 1 时，通信重新初始化， %MSG.E = 0 和 %MSG.D = 1
命令完成 输出	%MSG.D	状态为 1，表示命令完成，在下列情况下： • 发送成功则完成发送 • 完成接收（末尾字符接受） • 出错 • 功能块复位 状态为 0，请求处理
故障（出错） 输出	%MSG.E	状态为 1，如果发生下列情况： • 错误命令 • 不正确的配置表 • 接收到错误字符（通信速率，奇偶等错误） • 接收表已满（未更新） 状态为 0，报文长度，连接情况等都正常

当使用 EXCH 指令时发生错误，位 %MSG.D 和 %MSG.E 都会变为 1，且会在系统字 %SW69 内写入错误编码。见节 5.2-2。

输入复位 (R)：当输入复位为 1 时，它会立即中止任何正在发送的报文，把故障(Error)输出复位为 0 并把完成位置为 1。这样就可以发送一条新报文。

故障 (Error) 输出 (%MSG.E)：当发生通信编程错误或报文传送错误时，故障输出置为 **1**。如果与 EXCH 指令（字 **1**，低位有效字节）对应的数据块中定义的字节数超过了 **128**（十六进制中的 **80**），则故障输出置为 **1**。

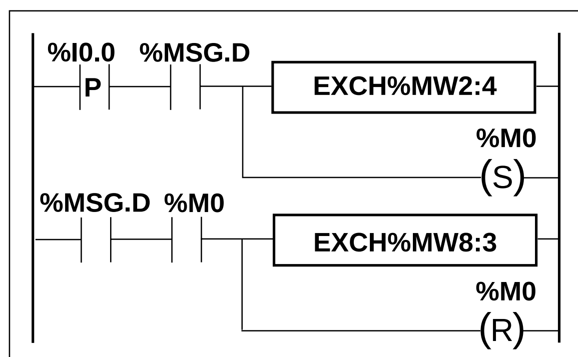
当发送一条 UNI-TELWAY 报文给 UNI-TELWAY 设备出现问题时，故障输出置为 1。在该情况下，用户应检查连线是否正确，目的设备是否支持 UNI-TELWAY 通信。

通信完成输出 (%MSG.D)：当完成输出为 **1** 时，**TSX Neza** 准备发送另一条报文。当发送多条报文时，最好使用 **%MSG.D** 位。如果不使用该位，可能导致报文丢失。

发送多条连续报文

执行 EXCH 指令会激活应用程序中的报文功能块。如果报文功能块未被激活 (%MSG.D=1),报文被发送。如果在一个周期内有多条报文被发送，只有第一条能被发送。用户可通过编程来实现多条报文的发送。

例如：连续发送两条报文。



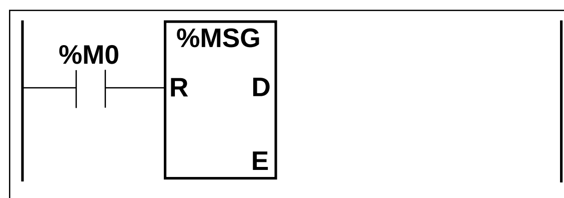

```
LDR     %I0.0
AND     %MSG.D
[EXCH  %MW2:4]
S       %M0
LD      %MSG.D
AND     %M0
[EXCH %MW8:3]
R       %M0
```

交换重新初始化

通过激活输入（或指令）R 来取消一次交换。该输入可初始化通信并把输出 %MSG.E 复位为 0，把输出 %MSG.D 复位为 1。如果检查到一个故障，可以重新初始化交换。

例如：BLK %MSG

```
LD %M0
R
END_BLK
```



特殊情况：

- 冷启动的影响：(%S0=1)强行重新初始化通信。
- 热启动的影响：(%S1=1) 无影响。
- **PLC** 停止的影响：如果一条报文正在发送，PLC 将中止发送并重新

初始化输出 %MSG.D 和 %MSG.E。

ASCII 模式的特性

可以通过终端端口来选择 ASCII 输出模式。当 TSX Neza 处于 ASCII 模式时，状态位 %S100 置为 1。

该指令的三个用途为：

- 发送
- 发送／接收
- 接收

发送或接收帧最大为 128 个字节。

与 EXCH 指令相关的字表由发送表和接收表组成。

高位字节	低位字节	
命令	长度 LgT/LgR	控制
发送的字节 1	发送的字节 2	发送表
...	...	
...	发送的字节 n	
发送的字节 n+1		
接收的字节 1	接收的字节 2	接收表
...	...	
...	接收的字节 p	
接收的字节 p+1		

长度字节表示发送字符的长度 (LgT)，在接收的结尾被改写为接收字符的长度 (LgR)。

命令字节必须包括下列值中的一个：

- 0: 发送
- 1: 接收后发送
- 2: 接收

当接收到帧的末尾字节后，接收结束 (1)。该字节的值可由用户修改（系统字 %SW68 的低位字节）。该字节的默认值为 H'0D'（回车）。

(1) 警告：
系统不能控制接收延时。

向一个 **ASCII** 设备发送一条报文：只传送

与 SEND 指令对应的数据块中需将数据写入一个 ASCII 设备（如显示器、打印机等）的内容有：

高位字节	低位字节
0（发送）	报文的长度（字节）
ASCII 数据	

报文的最大长度为 128 个字节。

例 **1**：要求使用 ASCII 协议在打印机上显示报文 "FAULT 10"。

%I0.0 %MSG.D
P — EXCH %KW10:7

```
LDR  %I0.0
AND  %MSG.D
[EXCH  %KW10:7]
```

字表的内容：

字	内容	高位字节	低位字节
%KW10	12	0	字节长度 LgT
%KW11	'DE'	ASCII 文本	
%KW12	'FA'		
%KW13	'UT'		
%KW14	'1'		
%KW15	'0'		
%KW16	16#0A0D	行回车	回车

ASCII 发送／接收

在发送结束时，TSX Neza 切换成等待接收的状态。如果接收状态正常且询问长度 (LgT) 和响应 (LgR) 的长度都小于 %MWi 保留区（长度为 L），那么等接收到响应后，把它复制到与发送表相关的 %MWi 区。如果不是这种情况，则位 %MSG.E 变为 1。
当检测到结束码或者接收表满了时，接收结束。(结束码的默认值为 H'0D'，但也可在 %SW68 内修改)。
在接收时无超时处理。

用于发送／接收 ASCII 数据，与 EXCH 指令相关的字表内容如下：

高位字节	低位字节	
1（发送／接收）	长度 LgT 或 LgR	控制
发送的字节 1	发送的字节 2	发送表
...	...	
...	发送的字节 n	
发送的字节 n+1		
接收的字节 1	接收的字节 2	接收表
...	...	
...	接收的字节 p	
接收的字节 p+1	结束码 (H'0D')	

注意

不能使用 %KWi 类字。

当交换结束（结束字符已接收），传送长度字节 LgT 包括接收字符数目(LgT)。
传送长度字节因而必须在每次交换前更新。

报文接收区被调整至发送区后的字。

例如：

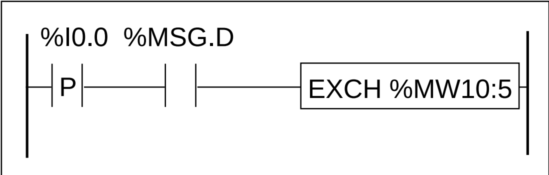
字	高位字节	低位字节
%MW10	16#0001	16#0007
%MW11	'V'	'A'
%MW12	'L'	'U'
%MW13	'E'	' '
%MW14	':'	未使用

相关程序：

LDR %I0.0

AND %MSG.D

[EXCH %MW10:5]



该程序发送下列帧：VALUE：7 个字节，等待接收响应（最大 8 个字节）。
可通过字 %MW15 到 %MW18 访问接收到的字符。

字表长度的计算

LgT = 7

LgR =8

$$L = 1 + \frac{LgT + LgT\%2}{2} + \frac{LgR + LgR\%2}{2} = 9$$

从 **ASCII** 设备接收报文

在执行已经被设置为接收的 EXCH 块时，TSX Neza 切换至等待接收的状态。如果接收状态正常且响应 (LgR) 的长度小于 %MWi 保留区（长度为 L），那么等接收到响应后，把它复制到与发送表相关的 %MWi 区。如果不是这种情况，则位 %MSG.E 变为 1。

当检测到结束码或者接收表满了时，接收结束。（结束码的默认值为 16#0D，但也可在 %SW68 内修改。）

在接收时无超时处理。

用于接收 ASCII 数据，与 EXCH 指令相关的字表内如下：

<table>
<tr><td>高位字节</td><td>低位字节</td><td></td></tr>
<tr><td>2（接收）</td><td>0 (1)</td><td>控制</td></tr>
<tr><td>接收的字节 1</td><td>接收的字节 2</td><td rowspan="4">接收表</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>...</td><td>接收的字节 P</td></tr>
<tr><td>接收的字节 p+1</td><td>结束码(H’0’D)</td></tr>
</table>

注意

> 不能使用 %KWi 类型字。

通信类型由表中的第 1 个字控制。

(1) 在接收模式中，没有考虑低位字节。

数据交换控制

使用 %MSG 功能块和系统字 %SW69 进行数据交换控制。

在每次交换后，都会更新 %SW69 中的内容（EXCH 块记录）：

- 0: 交换正常
- 1:发送表太大 (LgT>128)
- 2:发送表太小 (LgT=0)
- 3:字表太小 (1)
- 7:错误 ASCII 命令（命令类型 <> 0、1 或 2）
- 8:保留
- 9:接收错误（通信格式问题（速度、奇偶等））
- 10:在接收或发送／接收时不能使用 %KWi 表

$$L < 1 + \frac{LgT + LgT\%2}{2} + \frac{LgR + LgR\%2}{2}$$

L 用字表示，LgT 和 LgR 用字节表示。

与 **UNI-TELWAY** 设备对话

用于向一个 UNI-TELWAY 设备（如 ATV 速率控制器或者人机界面设备（CCX 17 或 XBT）等）发送请求，对应 EXCH 指令的字表由发送和接收表组成。

<table>
<tr><td>高位字节</td><td>低位字节</td><td></td></tr>
<tr><td>目标地址</td><td>长度 LgT / LgR（字节）</td><td>控制</td></tr>
<tr><td>分类码</td><td>请求码</td><td rowspan="4">发送表</td></tr>
<tr><td>第一个字 (MSB)</td><td>第一个字(LSB)</td></tr>
<tr><td>. . .</td><td>. . .</td></tr>
<tr><td>字 n-1 (MSB)</td><td>字 n-1 (LSB)</td></tr>
<tr><td>字 n (MSB)</td><td>字 n (LSB)</td><td rowspan="5">接收表</td></tr>
<tr><td>00（强制设置）</td><td>接收到的响应码</td></tr>
<tr><td>接收的数据 2</td><td>接收的数据 1</td></tr>
<tr><td>. . .</td><td></td></tr>
<tr><td>接收的数据 p-1</td><td>. . .
接收的数据 p</td></tr>
</table>

可发送和接收的最长报文为 128 字节。
报文接收区通常排在发送区后。

EXCH 块在发送长度区读取被发送的报文的长度 (LgT)。当交换结束时，写入接收到的报文的长度 (LgR)。
因此在每次发送前，发送长度字节都要被更新。

响应码写在接收表的第一个字的低位字节内。该字的高位字节被强置为 0。如果其后仍有数据，则写在下一个字内。

Uni-Telway 主站

在该模式中， TSX Neza 通常控制着两个设备，它们分散在 5 个从地址中。如果一个编程终端的地址是单一的，那么 TSX Neza 就可以控制两个设备和一个 PL707WIN 编程终端。

TSX Neza 不能管理从对从的链路。

TSX Neza Master 可以利用 EXCH 块把一条请求发送给 1 到 5 中的任意一个从地址。它使用源地址 0.254.16。

EXCH 块对应字表中的目标地址必须是下列值之一：

- 0:把一条请求发送到从地址 4
- 1 到 5:在 1 到 5 中的某一个从地址中发送和接收一条请求。

如果目标地址为 0，则 EXCH 块的特性如下：

- 接收缓冲未被使用，
- 表可位于 %KWi 区，
- 当从地址发出的响应被接收到时，位 %MSG.D 变为 1，
- 从地址发出的响应被忽略，
- 只有写请求和非请求数据请求可以使用。

如果目标地址是 1 到 5 中的一个：

- 必须有接收表（至少 1 个字），
- 字表必须位于 %MWi 区内，
- 当从地址发出的响应被接收到时，位 %MSG.D 变为 1，
- 从地址发出的响应被复制到接收表中。

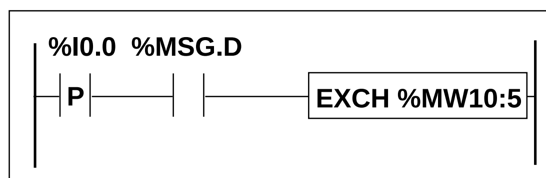
应用举例：

把请求“ Read word ”（读字） %MW510 (16#01FE) 发送到地址为 2 的设备中。

字	高位字节	低位字节
%MW10	02	04
%MW11	07	04
%MW12	01	FE

相关程序：

```
LDR  %I0.0
AND  %MSG.D
[EXCH %MW10:5]
```



EXCH 块使用 %MWi:L 作为参数：

- i 代表表内第一个字的地址
- L 代表字表内字的个数

当位 %MSG.D = 1 且位 %MSG.E = 0 时，表内包括以下数据：

字	高位字节	低位字节
%MW10	02	**04**
%MW11	07	04
%MW12	01	FE
%MW13	**00**	**34**
%MW14	**‘AB’**	**‘CD’**

粗体的数据表示：

- 接收到 4 个字节
- 接收到的响应码 =16#0034,
- 字 %MW510 的值 =16#ABCD

Unitelway 从站

任何设备（当地或远程）可以从 TSX Neza 从站（作为一个目标地址 Ad0）访问系统服务器。一个 TSX Neza 从站（客户）可以利用 EXCH 块向任一主或从设备（地址为 0 到 98）发送请求。EXCH 块对应字表中的目标地址应在 100 到 198 之间。

EXCH 块的特性如下：

- 必须有接收表（至少一个字），
- 字表必须位于 %MWi 区内，
- 当从地址发出的响应被接收到时，位 %MSG.D 变为 1，
- 从地址发出的响应被复制到接收表中。

应用举例：

把请求 “Read word”（读字）%MW510 发送到主机中。

字	高位字节	低位字节
%MW10	00	04
%MW11	07	04
%MW12	01	FE

相关程序：

```
LDR     %I0.0
AND     %MSG.D
[EXCH   %MW10:5]
```

EXCH 块使用 %MWi:L 作为参数：

- i 代表表内第一个字的地址
- L 代表字表内字的个数

当位 %MSG.D = 1 且位 %MSG.E = 0 时，表内包括以下数据：

字	高位有效字节	高位有效字节
%MW10	00	**04**
%MW11	07	04
%MW12	01	FE
%MW13	**00**	**34**
%MW14	**'AB'**	**'CD'**

粗体的数据表示:

- 接收到 4 个字节,
- 接收到的响应码 = 16#0034,
- 字 %MW510 的值 = 16#ABCD

其他例子:

把请求“ Read word ” %MW510 发送到从 32 中。

字	高位字节	低位字节
%MW10	20	04
%MW11	07	04
%MW12	01	FE

相关程序:

LDR%I0.0

AND%MSG.D

[EXCH%MW10:5]

EXCH块使用 %MW:L 作为参数 :

- i 代表表内第一个字的地址
- L 代表字表内字的个数

当 %MSG.D = 1 并且 %MSG.E = 0 时,表内包括以下数据 :

字	高位字节	低位字节
%MW10	20	**04**
%MW11	07	04
%MW12	01	FE
%MW13	**00**	**34**
%MW14	**'AB'**	**'CD'**

粗体的数据表示:

- 接收到 4 个字节,
- 接收到的响应码 = 16#0034,
- 字 %MW510 的值 = 16#ABCD

数据交换控制

使用 %MSG 功能块和系统字 %SW69 进行数据交换控制。

在下列情况下位 %MSG.D 设为 1：

- 接收到响应时
- 出现发送错误（接收失败 NAK）
- 功能块被重新初始化
- 7 秒内响应没有到达 （响应超时）

出现其它任何错误（在字 %SW69 中表述），位 %MSG.D 都变为 1：

在每次交换后，%SW69 中的内容（EXCH 块记录）更新为：

- 0: 交换正常
- 1: 发送表太大 (LgT>128)
- 2: 发送表太小 (LgT=0)
- 3: 字表太小 (1)
- 4: Unitelway 地址错误

在 UNI-TELWAY 从模式中目标地址不在 [0...98] 或 [100...198] 内

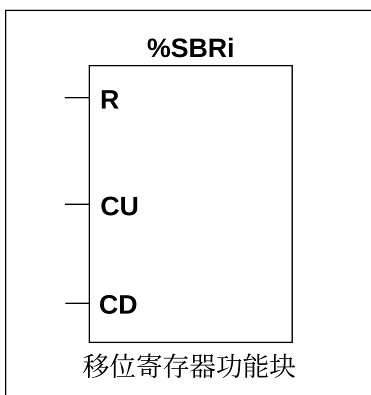
在 UNI-TELWAY 主模式中目标地址不在 [1...5] 内

- 5: 超时中止
- 6: 发送错误（对方响应 NAK）
- 7: 错误 ASCII 命令（命令字节 < > 0、1 或 2）
- 8: 保留
- 9: 接收错误（通信格式问题（通信速度、奇偶等））
- 10: 在接收或发送／接收时使用了被禁止的 %KWi 表

(1) $L < 1 + \frac{LgT + LgT\%2}{2} + \frac{LgR + LgR\%2}{2}$

L 用字表示，LgT 和 LgR 用字节表示。

3.2-7 移位寄存器功能块 %SBRi



移位寄存器功能块

移位寄存器用于存放二进制数据（0 或 1），
并把它们向一个方向移动。

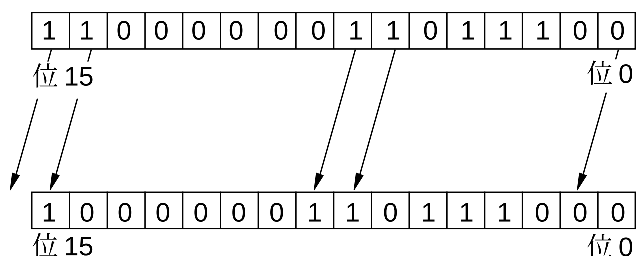
特性		
寄存器编号	%SBRi	0 到 7
寄存器位	%SBRi.j	寄存器中的 0 到 15 位（j = 0 到 5） 可由测试指令进行测试，由赋值指令写入值
输入复位 （或指令）	R	在一个上升沿处，把寄存器位 %SBRi.j 0 到 15 设为 0
左移位输入 （或指令）	CU	在上升沿处，把寄存器位向左移
右移位输入 （或指令）	CD	在上升沿处，把寄存器位向右移

操作

初始状态

CU %SBRi 执行向左移位

位 15 丢失

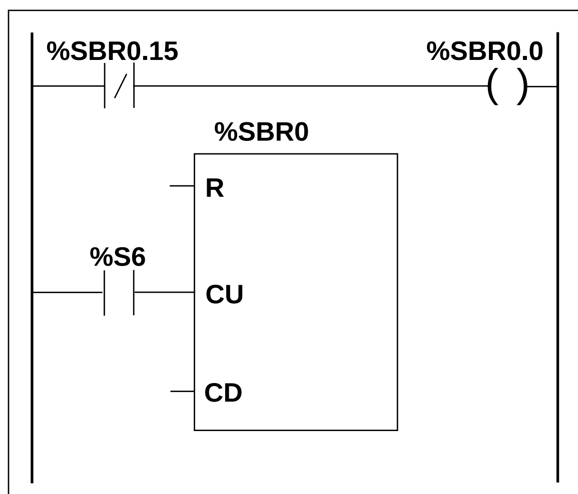


使用指令 CD 也可右移 1 位，位 0 丢失。

如果 16 位寄存器不够用，则通过编程串联几个寄存器。

编程

例如：每秒向左移动一位：假设位 0 与位 15 的状态相反。



可逆编程

```
LDN     %SBR0.15
ST      %SBR0.0
BLK     %SBR0
LD      %S6
CU
END_BLK
```

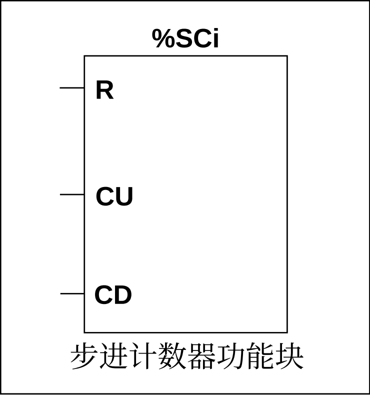
不可逆编程

```
LDN     %SBR0.15
ST      %SBR0.0
LD      %S6
CU      %SBR0
```

特殊情况:

- 冷启动的影响 :(%S0=1)
  - 把所有的寄存器字的位置为 0
- 热启动的影响 :(%S1=1) 对寄存器字内的位无影响。

3.2-8 步进计数器功能块 %SCi

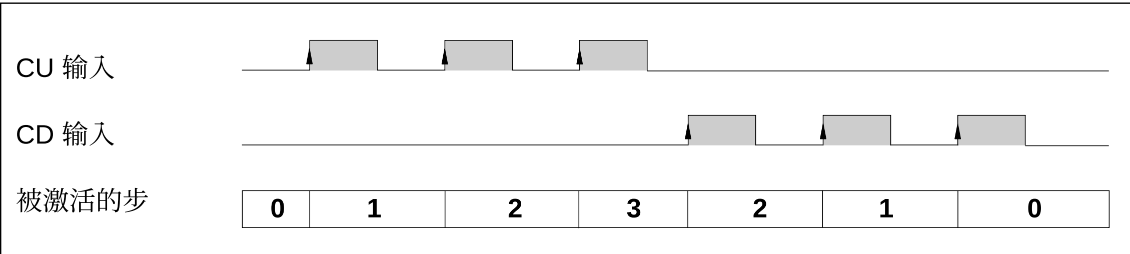


步进计数器功能块

步进计数器由一系列动作可被赋值的步组成。由外部或内部事件决定从一步移到另一步。当一步被激活时，其相应位置为 1。步进计数器中每次只能有一步被激活。

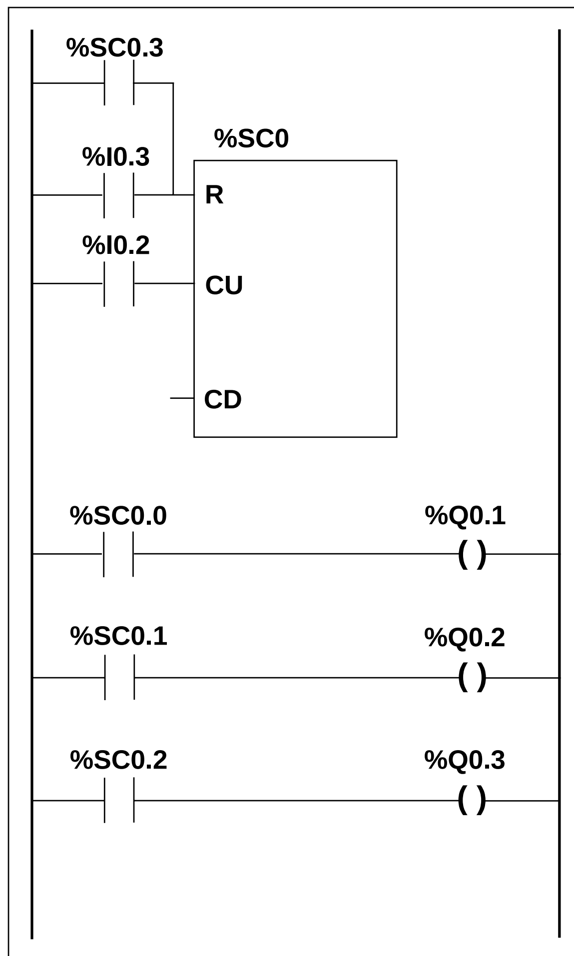
特性		
步进计数器编号	%SCi	0 到 7
步进计数器位	%SCi.j	寄存器中的 0 到 225 位(j = 0 到 225) 可由 LD 逻辑操作进行测试，由赋值指令写入值
复位输入 （或指令）	R	在上升沿，复位步进计数器
递增输入 （或指令）	CU	在上升沿，递增步进计数器一步
递减输入 （或指令）	CD	在上升沿，递减步进计数器一步

操作



编程

例如：通过输入 %I0.2 递增步进计数器 0。通过输入 %I0.3 或者当它达到步 3 时把它复位为 0。步 0 控制输出 %Q0.1，步 1 控制输出 %Q0.2，步 2 控制输出 %Q0.3。



可逆编程

```
BLK     %SC0
LD      %SC0.3
OR      %I0.3
R
LD      %I0.2
CU
END_BLK
LD      %SC0.0
ST      %Q0.1
LD      %SC0.1
ST      %Q0.2
LD      %SC0.2
ST      %Q0.3
```

不可逆编程

```
LD      %SC0.3
OR      %I0.3
R       %SC0
LD      %I0.2
CU      %SC0
LD      %SC0.0
ST      %Q0.1
LD      %SC0.1
ST      %Q0.2
LD      %SC0.2
ST      %Q0.3
```

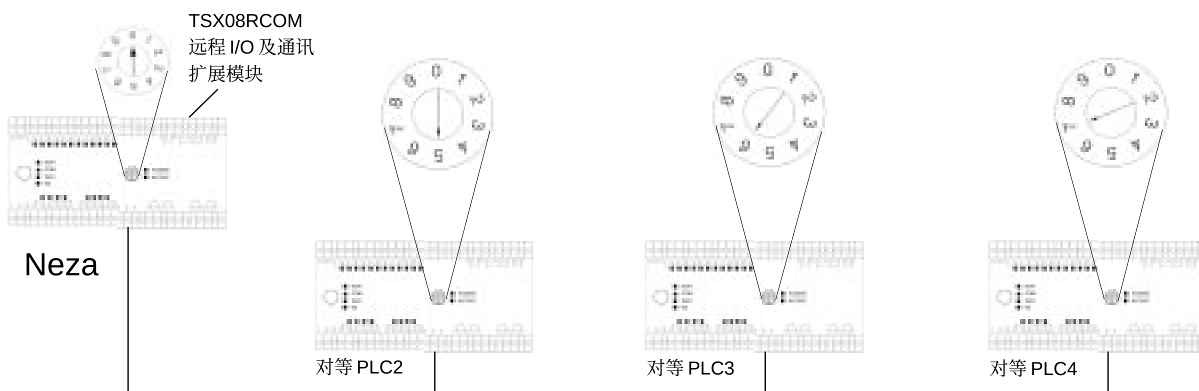
特殊情况

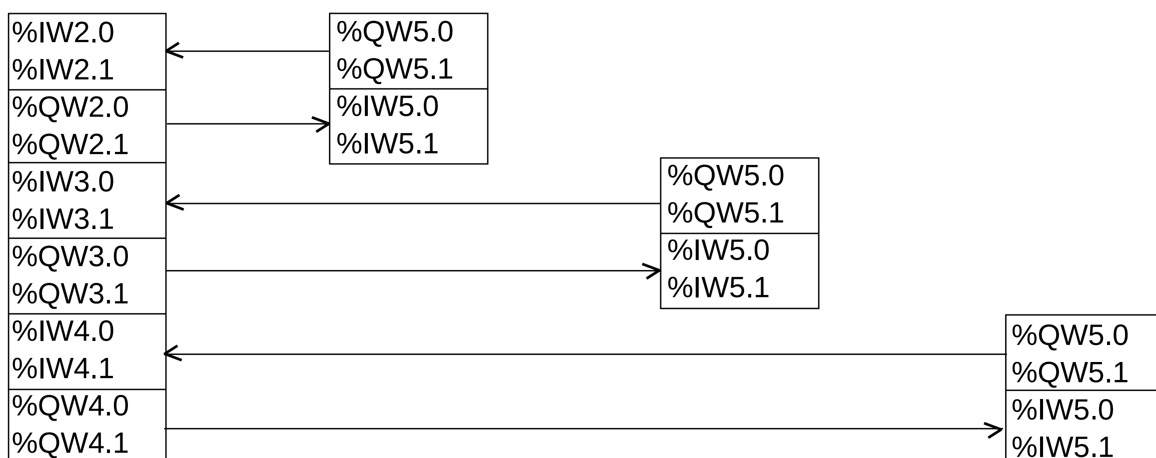
- 冷启动的影响：(%S0=1)
  - 初始化步进计数器
- 热启动的影响：(%S1=1) 对步进计数器无影响

3.3 PLC 间的通信

对等 PLC 间的通信

字 %IW 和 %QW 用于在 PLC 之间交换数据。下图列出了每个 PLC 的交换字。当 PLC 处于 RUN 模式时，这些交换字自动更新。





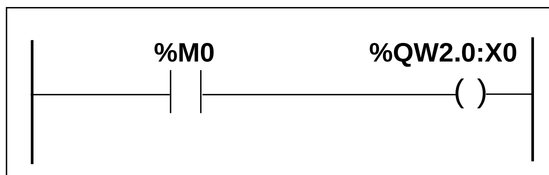
每个 PLC 的用户程序限于：

- 写输出字 %QWi.j
- 读输入字 %IWi.j

字 %IW/%QW 的更新周期与 PLC 扫描同步。当一个完整的更新周期完成后，系统位 %S70 置为 1。它可由程序或终端复位为 0。位 %S71 / %S72 和字 %SW71 也可用于控制数据交换。

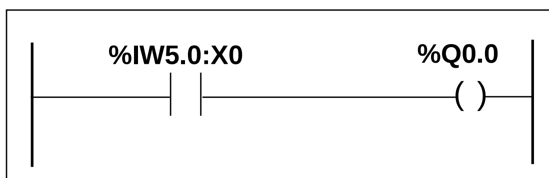
注意：每个 PLC 地址都是根据远程 I/O 及通讯扩展模块 TSX08RCOM 上的旋转拨码开关的位置决定的，并且在每一次上电时该位置起效。

例 1:主 PLC 向对等 PLC #2 发送数据表示生产的结束（位 %M0=1）。一旦对等 PLC 接收到该数据，它就通过激活输出 %Q0.0 来启动一台物料处理机器。



本体 PLC 编程

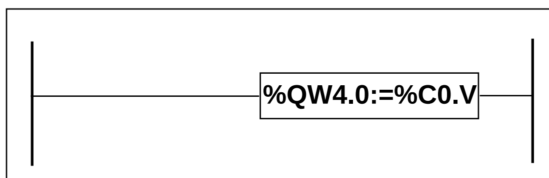
```
LD      %M0
ST      %QW2.0:X0
```



对等 PLC 编程

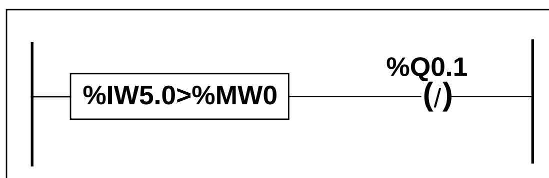
```
LD      %IW5.0:X0
ST      %Q0.0
```

例 2:主 PLC 向对等 PLC #4 发送计数器 0 的当前值。若当前值大于字 %MW0 所表示的阈值时，对等 PLC 通过中止输出 %Q0.1 来停止机器。



本体 PLC 编程

```
LD      1
[%QW4.0:=%C0.V]
```



对等 PLC 编程

```
LD      [%IW5.0 >%MW0]
STN     %Q0.1
```

Block I/O 扩展**PLC与主 PLC** 间的通讯

以 Block I/O 扩展连接的 PLC，其所连接的远程 I/O 及通风扩展模块上的旋转拨码开关可在位置 1~4，受主 PLC 的控制，其输入地址为 %Ii.j(i=1-4, j=0-11)，输出地址为 %Qi.j(i=1-4,j=0-7)。

4 时钟功能

4.1 简介

带实时钟的 Neza 有实时时钟功能，它可以实现三个功能：

- 调度模块，用于控制动作，这些动作预先根据时钟定义。
- 时间／日期标记，用于记录事件发生的日期并测量持续的时间。

TSX Neza 的调度模块和实时时钟可以从 TSX PLC 的操作菜单中得到。另外，实时时钟也可以由程序来设置。如果在 PLC 断电之前对其电池连续充电 6 个小时以上，那么在 PLC 掉电之后，调度模块和日历时钟还可以连续运行 20 天。

日历时钟为 24 小时格式，并且可以考虑到闰年情况。

4.2 调度模块

调度模块用于控制在特定预定时间执行的动作。最多可以有 16 个调度模块。这些模块不需要编程就可以对它们进行配置。

4.2-1 特性

调度模块序号	**RTC : n**	n = 0 到 15
输出	**Q :**	输出任务，激活 %Mi 或 %Qj.k 。 当当前日期和时间在设定的激活期内时， 该输出端置 1。
开始日期	**DD:MMM**	显示调度模块激活的起始日 1 到 31 和起始月 1 月到 12 月。
结束日期	**DD:MMM**	显示调度模块激活的结束日 1 到 31 和结束月 1 月到 12 月。
星期	**MTWTFSS**	显示一周内激活的日期。 （M:星期一 到 S:星期日）
开始时间	**hh:mm**	显示调度模块激活的起始时（0 到 23）和分（0 到 59）。
结束时间	**hh:mm**	显示调度模块激活的结束时（0 到 23）和分（0 到 59）。

B

系统字 %SW114 的位可以使每一个功能块的操作使能（相应位置 1）或禁止（相应位置 0）。

%SW114

调度模块 #15　　　　　　调度模块 #0

默认情况下（或冷启动以后），这个系统字的所有位都置为 1。程序可以根据需要使用这些位。

注释：

- 如果几个调度模块赋值经同一个输出（%Mi 或 %Qj.k），则最后赋给它的值是每个调度模块的值的 “ OR ”(也就是说，同一个输出有可能有好几个 “ 操作范围 ”)。

示例：为夏季喷洒程序配置调度模块。

- RTC **6** :调度模块 #6
- Q : **%Q0.2** :调度模块激活输出
- **21 -Jun -> 21-Sep** :激活期间（6 月 21 日 **->** 9 月 21 日）
- 星期一、星期三和星期五激活日
- **21 : 00 - 22 : 00** :计划激活时间

RTC:6　Q:%Q0.2
21 - Jun -> 21 - Sep
星期一，星期三和星期五
21 : 00 - 22 : 00

%I0.1　**%SW114 : X6**
( )

```
LD      %I0.1
ST      %SW114:X6
```

%I0.1
21 June　**21 Sep**
%Q0.2
星期 一 三 五　一 三 五　一 三 五

本例中，用户可以用一个开关或湿度检测器连接到输入端 %I0.1 禁止该调度模块。

注意：

检查 %S51 位的状态是很重要的，它给出了调度模块的错误信息。

4.2-2 时钟在程序中的使用

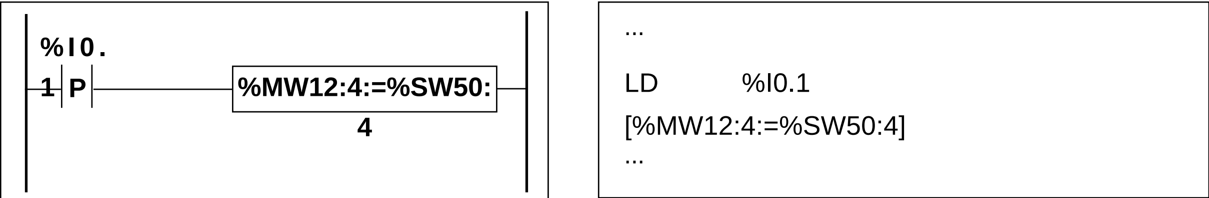
日期和时间可以从系统字 %SW50 到 %SW53（参见第 5.2 节）中得到。因此，由 PLC 程序可修改日期和时间的当前值，或与包含给定值的字 %MWi （或 %KWi）进行算术比较，就可以实现时间和日期的标记。

4.3 时间／日期标记

系统字 %SW50 到 %SW53（参见第 5.2 节）中包含 BCD 码格式的当前日期和时间，这对于在外设上的显示与传送是十分有用的。这些系统字还可以用于存储事件的日期和时间。

要标记一个事件的日期，常见的操作是 : 利用赋值操作，将系统字的内容传送到内部字中，然后再处理这些内部字（例如，用 EXCH 指令传送到显示单元）。

示例：



一旦检测到一个事件，字表将包含以下内容：

编码	高位有效字节	低位有效字节
%MW12	秒	星期 (1)
%MW13	时	分
%MW14	月	日
%MW15	世纪	年

举例:	1994 年 4 月 19 日，星期一
十六进制	13 点 40 分 30 秒
3000	30 秒，0 = 星期一
1340	13 时，40 分
0419	4 = 四月，19 日
1994	1994 年

(1) 其中 0 = 星期一，1 = 星期二，2 = 星期三，3 = 星期四，4 = 星期五，5 = 星期六，6 = 星期日。

使用系统字读取 **PLC** 上一次停止的日期和时间

系统字 %SW54 到 %SW57（参见第 5.2 节）记录 PLC 上一次的日期和时间，系统字 %SW58 包含了导致上一次停止的原因的代码，均为 BCD 格式。

4.4 设置日期和时间

4.4-1 使用编程终端设置日期和时间

当编程终端在 TSX Neza（在线）模式时，可以简单而快速地更改日期和时间（参见 C 部分的 5.18 节）。

4.4-2 使用系统字更新日期和时间

系统字提供了另外两种更新日期和时间的可能方式：

使用系统字 %SW50 到 %SW53 更新（参见 5.2 节）

为此，%S50 位必须设置为 1。这个位可以：

- 取消内部时钟（字 %SW50 到 %SW53 ）的更新；
- 将字 %SW50 到 %SW53 中的值传送到内部时钟。

%S50
%S50 (R)
%I0.1 P
[%SW50:=%MW10]
[%SW51:=%MW11]
[%SW52:=%MW12]
[%SW53:=%MW13]
%S50 (S)

```
LD      %S50
R       %S50
LDR     %I0.1
[%SW50: = %MW10]
[%SW51: = %MW11]
[%SW52: = %MW12]
[%SW53: = %MW13]
S       %S50
```

字 %MW10 到 %MW13 中包含 BCD 码格式的最新日期和时间。

字表一定包含最新的日期和时间：

编码	高位字节	低位字节
%MW10	秒	星期一
%MW11	时	分
%MW12	月	日
%MW13	世纪	年

举例：	2000 年 4 月 19 日，星期三
十六进制	13 点 40 分 30 秒
3002	30 秒，2 = 星期三
1340	13 时，40 分
0419	4 = 四月，19 日
2000	2000 年

(1) 其中 0 = 星期一，1 = 星期二，2 = 星期三，3 = 星期四，4 = 星期五，5 = 星期六，6 = 星期日。

使用系统字 %SW59 更新

更新日期和时间的另一种方法是使用使能位 %S59 和调整字 %SW59。将 %S59 位设置为 1，这样字 %SW59 就可以调整当前日期和时间。关于字 %SW59 的叙述在第 5.2 节。在上升沿分别增加或减小日期和时间。

例如：建立一个面板，用于修改内部时钟的时、分和秒。



%M0		%S59
%I0.2	%I0.0 P	%SW59:X3
%I0.2	%I0.1 P	%SW59:X11
%I0.3	%I0.0 P	%SW59:X2
%I0.3	%I0.1 P	%SW59:X10
%I0.4	%I0.0 P	%SW59:X1
%I0.4	%I0.1 P	%SW59:X9

```
LD      %M0
ST      %S59
LD      %I0.2           时
ANDR    %I0.0
ST      %SW59:X3
LD      %I0.2
ANDR    %I0.1
ST      %SW59:X11
LD      %I0.3           分
ANDR    %I0.0
ST      %SW59:X2
LD      %I0.3
ANDR    %I0.1
ST      %SW59:X10
LD      %I0.4           秒
ANDR    %I0.0
ST      %SW59:X1
LD      %I0.4
ANDR    %I0.1
ST      %SW59:X9
```

- 输入 %I0.2、%I0.3 和 %I0.4 为时／分／秒开关控制。
- 输入 %I0.0 执行增加，+ 按钮。
- 输入 %I0.1 执行减小，- 按钮。

5 系统位和系统字

5.1 系统位

5.1-1 系统位列表

位	功能	初始状态	控制
%S0	1 = 冷启动（电源恢复时数据丢失）	0	S 或 U->S
%S1	1 = 热启动（电源恢复时数据不丢失）	0	S 或 U->S
%S4, %S5	时基 10ms、100ms	-	S
%S6, %S7	时基 1s、1min	-	S
%S8	0 = 当 PLC 停止时输出保持	1	U
%S9	1 = 如果 PLC 运行，则 PLC 的输出复位为 0	0	U
%S10	0 = I/O 故障	1	S
%S11	1 = 警戒时钟溢出	-	S
%S13	1 = PLC RUN 后的第一次扫描	1	S
%S17	1 = 无符号算术运算或循环移位溢出	0	S->U
%S18	1 = 算术运算溢出或出错	0	S->U
%S19	1 = 扫描周期超时	0	S->U
%S20	1 = 索引溢出	0	S->U
%S50	1 = 设置调度模块 (RTC)	0	U
%S51	1 = 调度模块没有初始化或出错 0 = 设置调度模块	0	S
%S59	1 = 调整当前日期	0	U
%S70	1 = 更新扩展上的交换字 %IW/QW。处理 Modbus 请求	0	S
%S71	1 = 通过扩展连接交换	0	S
%S72	0 = 对等 PLC 的扫描	0	U
%S100	/DPT 的状态	-	S
%S101	通讯端口设置	0	0
%S118	1 = 主 PLC 故障	0	S
%S119	1 = 对等 PLC 故障	0	S

S = 系统控制；U = 用户控制；U->S = 由用户置 1，系统复位为 0；S->U = 由系统置 1，用户复位为 0。

5.1-2 系统位的详细说明

TSX Neza PLC 的系统位 %Si 显示了 PLC 的状态，或者使用用户的干预被执行。

可以在用户程序中检查这些位，以便检测任何需要特殊处理的事件。有一些位必须由程序复位到初始或正常状态，然而，已经由系统复位到初始或正常状态的系统位不要再用程序或终端复位。

系统位	功能	描述
%S0	冷启动	正常值为 0，置 1 时： • 电源恢复，数据丢失（电池故障）； • 用户程序； • 终端（在数据编辑器中）。 该位在第一次完全扫描过程中置 1，并在下一次扫描之前复位为 0。 关于操作信息：参见 A 部分的第 7.1 节。
%S1	热启动	正常值为 0，置 1 时： • 电源恢复，并且保存数据； • 用户程序； • 终端（在数据编辑器中）。 该位在第一次完全扫描结束且在更新输出之前由系统复位为 0。 关于操作信息：参见 A 部分的第 7.1 节。
 %S4 **%S5** **%S6** **%S7**	时基 10ms 100ms 1s 1min	这些位状态的改变由一个内部时钟控制，而且不与 PLC 的扫描同步。 例如：%S4 5ms 5ms
%S8	输出保持	初始值为 1，可以由程序或终端（在数据编辑器中）置 0： • 在状态 1 时，如果程序没有被正常执行或者 PLC 停止时 PLC 输出为 0； • 在状态 0 时，如果程序操作出错或者 PLC 停止时 PLC 输出保持当前状态。
%S9	输出复位	正常值为 0，可以由程序或终端（在数据编辑器中）置 1： • 在状态 1 时，当 PLC 在 RUN 模式时 PLC 输出强置为 0； • 在状态 0 时，PLC 输出被正常刷新。

系统位	功能	描述
%S10	I/O 故障	正常值为 1，当检测到主 PLC 或对等 PLC 上的 I/O 故障（配置故障、交换故障、硬件故障）时，该位被置 0。%SW118 和 %SW119 位显示故障在哪一个 PLC 上 %SW118 和 %SW119 字给出故障的详细信息(参见 5.2 节)。 当故障排除时，%S10 位复位为 1。
%S11	警戒时钟溢出	正常值为 0，当程序执行时间（扫描时间）超过最大扫描时间（软件警戒时钟）时，该位由系统置 1。警戒时钟溢出将导致 PLC 变为 STOP 状态。
%S13	第一次扫描	正常值为 0，在 PLC 变为 RUN 之后的第一次扫描过程中，该位由系统置 1。
%S17	进位溢出	正常值为 0，以下情况将被系统置 1： • 当无符号的算术运算（余数）进位溢出时； • 在循环或移位操作过程中，它表示 1 被移出。 在有溢出可能的地方，用户程序必须在每一次操作之后检查该位是否有溢出危险，当溢出发生时，用户要将其复位为 0。
%S18	算术运算 溢出或出错	正常值为 0，在执行 16 位运算溢出时置 1，即： • 运算结果大于 + 32767 或小于 - 32768； • 0 作除数； • 负数求平方根； • BTI 或 ITB 转换无意义：BCD 码的值超出范围。 在有溢出可能的地方，用户程序必须在每一次操作之后检查该位，当有溢出发生时，用户要将其复位为 0。
%S19	扫描时间 超限（周 期扫描）	正常值为 0，当扫描时间超限（扫描时间大于用户在配置或在 %SW0 中设定的时间）时由系统置 1。 此位由用户复位为 0。
%S20	索引溢出	正常值为 0，当索引对象的地址小于 0 或大于最大值时，该位被置 1。 在有溢出可能的地方，用户程序必须在每一次操作之后检查该位，当有溢出发生时，将其复位为 0。

系统位	功能	描述
%S50	使用字 %SW50 到 53 更新日期和时间	正常值为 0，这个位可以由程序或终端置 1 或置 0。 • 为 0 时，日期和时间可以读出 • 为 1 时，日期和时间可以被更新。
%S51	实时时钟状态	• 为 0 时，日期和时间已经设置好。 • 为 1 时，日期和时间必须由用户来设置。 当这个位为 1 时，实时时钟数据为无效状态。此时，日期和时间可能未被设置，或者电池电压太低。
%S59	使用字 %SW59 更新日期时间	正常值为 0，这个位可以由程序或终端置 1 或置 0。 • 为 0 时，日期和时间保持不变。 • 为 1 时，日期和时间根据 %SW59 中设置的控制位增加或减少。
%S70	更新交换字 处理 Modbus 请求	对于主 PLC 来说，当完成一次传送交换字 %IW/%QW 到对等 PLC 的完整周期时，该位就置为 1。 对于每一个对等 PLC，当对等 PLC 与主 PLC 完成接收并传送交换字时，该位就置为 1。 该位由程序或终端复位为 0。 当一个 Modbus 请求被处理时，该位就置 1。 操作员可以使用这个位。 该位由程序或编程终端复位为 0。
%S71	通过扩展连接进行交换	初始值为 0。当检测到一个通过扩展连接的交换时，该位置 1。当没有通过扩展连接执行交换时，该位置 0。主 PLC 的字 %SW71 显示了有效扩展的清单和状态。
%S100	/DPT 信号的状态	显示 TER 端口上的 INL/DPT 短接状态： • 未短接：UNI-TELWAY 主协议（%S100 = 0） • 短接状态：（/DPT 为 0V）协议由应用程序的配置（%S100 = 1）来定义
%S101	通讯端口设置	当 %S101=0，（缺省）时，由 EXCH 指令控制的通讯数据由 TER 口发送 / 接收。 当 %S101=1，由 EXCH 指令控制的通讯数据由扩展通讯口发送 / 接收。
%S118	主 PLC 故障	正常值为 0，当检测到主 PLC 上的 I/O 故障时置 1。字 %SW118 给出了故障的详细内容。 当故障消失时，位 %S118 复位为 0。
%S119	对等 PLC 故障	正常值为 0，当检测到 I/O 扩展上的 I/O 故障时置 1。字 %SW119 给出了故障的详细内容。 当故障消失时，位 %S119 复位为 0。

5.2 系统字

5.2-1 系统字列表

字	功能	控制
%SW0	PLC 扫描周期值	U
%SW11	软件警戒时钟	S
%SW14	UNITELWAY 超时	S 和 U
%SW15	PLC 版本和 UI	S
%SW30	PLC 最后一次扫描周期的时间	S
%SW31	PLC 扫描周期的最大时间	S
%SW32	PLC 扫描周期的最小时间	S
%SW50 %SW51 %SW52 %SW53	调度模块 (RTC) 功能：该字包含当前日期和时间值（BCD 码格式） %SW50 = 秒和星期 %SW51 = 时和分 %SW52 = 月和日 %SW53 = 世纪和年	S 和 U
%SW54 %SW55 %SW56 %SW57	调度模块 (RTC) 功能：该字包含上一次电源故障和 PLC 停止的日期和时间值（BCD 码格式） %SW54 = 秒和星期 %SW55 = 时和分 %SW56 = 月和日 %SW57 = 世纪和年	S
%SW58	上一次停止的标识码	S
%SW59	调整当前日期	U
%SW67	Modbus 帧结束字符值（ASCII 模式）	U
%SW68	接收的帧结束字符值（ASCII 模式）（TER 端口）	U
%SW69	EXCH 模块出错代码	S
%SW70	TSX 08 Neza PLC 的地址	S
%SW71	扩展连接上的设备	S
%SW76	1 ms 定时器	S
%SW77	1 ms 定时器	S
%SW78	1 ms 定时器	S
%SW79	1 ms 定时器	S
%SW100	模拟量模块输入功能命令字	U
%SW101	采集到的模拟量模块输入值	S
%SW102	模拟量模块输出功能命令字	U
%SW103	要生成的模拟量模块输出值	U
%SW110	读取计数器的值	S
%SW111	高速计数功能	S 和 U
%SW114	调度模块 (RTC) 使能	U
%SW116	模拟量模块 (EA4A2) 设置	U
%SW117	模拟量模块 (EAV8A2 / EAP8) 设置	U
%SW118	主 PLC 状态字	S
%SW119	对等 PLC 状态字	S

S= 由系统控制　　U= 由用户控制

5.2-2 系统字的详细说明

TSX Neza PLC 使用以下系统字：

系统字	功能	描述
%SW0	PLC 扫描周期	通过用户程序或编程终端（在数据编辑器中）修改在配置中定义的 PLC 扫描周期。
%SW11	警戒时钟时间	读取警戒时钟时间 (150ms)。
%SW14	Unitelway 超时	用于通过用户程序修改 UNITELWAY 超时的值（参见通讯部分的 7.3-3 节）。
%SW15	PLC 版本和 UI	该字用于显示 PLC 的版本（高位字节）和它的 UI （低位字节）。 例如：TSX08 Neza 为 0X1019: PLC 的版本为 1.0, UI 为 27。
%SW30	上一次扫描时间 (1)	显示 PLC 上一次扫描的执行时间（以 ms 为单位）。
%SW31	最大扫描时间	显示 PLC 上一次冷启动后最长的扫描执行时间(以 ms 为单位)。
%SW32	最小扫描时间（1）	显示 PLC 上一次冷启动后最短的扫描执行时间(以 ms 为单位)。
%SW50 %SW51 %SW52 %SW53	实时时钟	包含当前日期和时间（BCD 码方式）的系统字： %SW50 ： SSXN 秒和星期（N = 0 为星期一到 6 为星期日） %SW51 : HHMM 时和分 %SW52 : MMDD 月和日 %SW53 : CCYY 世纪和年 当位 %S50 为 0 时，这些字由系统控制。当位 %S50 为 1 时，这些字可由用户程序或编程终端写入。
%SW54 %SW55 %SW56 %SW57	上次停机时间	该系统字包含上一次电源故障或 PLC 停止的日期和时间值（BCD 码方式）： %SW54 = 秒和星期 %SW55 = 时和分 %SW56 = 月和日 %SW57 = 世纪和年。

(1) 这个时间与扫描周期的开始（采集输入）到结束（更新输出）之间的时间对应。

<table>
<tr><th>系统位</th><th>功能</th><th colspan="3">描述</th></tr>
<tr><td>%SW58</td><td>上一次停止的标记码</td><td colspan="3">显示导致上一次停止的代码:
1 = 终端开关从 RUN 变为 STOP
2 = 软件故障导致停止（PLC 扫描过长）
4 = 停电
5 = 硬件故障导致停止</td></tr>
<tr><td rowspan="3">%SW59</td><td rowspan="3">调整当前实时时钟</td><td colspan="3">包含两组 8 位，用于调整当前的日期和时间。操作总是在位的上升沿执行，该字由 %S59 位使能。</td></tr>
<tr><td>增加</td><td>减小</td><td>参数</td></tr>
<tr><td>第 0 位
第 1 位
第 2 位
第 3 位
第 4 位
第 5 位
第 6 位
第 7 位</td><td>第 8 位
第 9 位
第 10 位
第 11 位
第 12 位
第 13 位
第 14 位
第 15 位</td><td>星期
秒
分
时
日
月
年
世纪</td></tr>
<tr><td>%SW67</td><td>Modbus 帧结束代码</td><td colspan="3">用于在 Modbus 的结束帧设置 ‘ LF ’（ASCII 模式）。
在冷启动时，该字由系统写为 16#000A。当主 PLC 使用的帧结束字符不是 16#000A 时, 用户可以使用程序或调整模式修改这个字。</td></tr>
<tr><td>%SW68</td><td>接收的帧结束代码（ASCII 模式）</td><td colspan="3">用于设置帧结束的参数（ASCII 模式)。一收到这个值立即停止接收。
默认值为：16#000D。</td></tr>
<tr><td>%SW69</td><td>EXCH 模块出错代码</td><td colspan="3">在使用 EXCH 块出错时, 输出位 %MSG.D 和 %MSG.E 变为 1。这个系统字包含出错的代码，其值如下：
0:无错误，交换正确
1:传送缓冲区太大
2:传送缓冲区太小
3:表太小
4:错误的 Unitelway 地址（仅在 Unitelway 模式）
5:超时（Unitelway 模式或 Modbus 主模式）
6:传送错误（仅在 Unitelway 模式）
7:错误 ASCII 命令（仅在 ASCII 模式）
8: 保留
9:接收错误（仅在 ASCII 模式）
10:%KWi 字表禁止
20:Modbus 从地址错误（在 Modbus Master 模式下）</td></tr>
</table>

系统位	功能	描述
%SW69 （续）		21:Neza 不支持的 Modbus 功能码（仅在 Modbus Master 模式） 22:重试次数无效（有效值为 0-3，仅在 Modbus Master 模式） 23:数据长度无效(Modbus Master 模式） 24:结束参数号(开始参数号+长度)无效(在 Modbus Master 模式下) 81:从 PLC 返回“ 非法功能码 ”信息 82:从 PLC 返回“ 非法数据地址 ”信息 83:从 PLC 返回“ 非法数据值 ”信息 84:从 PLC 返回“ 从设备失败 ”信息 85:从 PLC 返回“ 确认 ”信息 86:从 PLC 返回“ 从设备忙 ”信息 87:从 PLC 返回“ 未确认 ”信息 88:从 PLC 返回“ 内存奇偶校验错误 ”信息 每次使用 EXCH 块后，该字被清 0。
%SW70	PLC 地址	包含如下信息： • 第 2 位：1 = 有调度模块 (RTC) • 第 7、6、5 位：PLC 的地址（和 TSX08RCOM 上的旋转拨码开关位置相同） 如果有 I/O 扩展： • 第 13 位：1 = 有一个 I/O 扩展
%SW71	远程扩展连接上的设备	显示每一个远程扩展与主 PLC 的通讯状态： 第 1 位： Block I/O 扩展 第 2 位：对等 PLC 或模拟量模块 #2 第 3 位：对等 PLC 或模拟量模块 #3 第 4 位：对等 PLC 或模拟量模块 #4 相应位为 0：如果没有远程扩展或对等 PLC, 没有供电或有故障。 相应位为 1：如果有远程扩展并与主 PLC 交换数据。
%SW76 到 %SW79	减计数字 1 ms	这 4 个字用作 1 ms 定时器。如果它们的值为正，则每毫秒由系统分别减 1。这就构成了 4 个毫秒减计数器，相当于操作范围为：1 ms 到 32767 ms。设置第 15 位为 1 可以停止减操作。
%SW100	模拟量输入	模拟量输入功能命令字。 值为 0: 模拟量输入无效 值为 1: 无量程操作 值为 2: 单极量程 (周期 125ms) 值为 3: 双极量程 (周期 125ms) 值为 4: 单极量程 (周期 500ms) 值为 5: 双极量程 (周期 500ms) 该字必须由应用程序进行写操作。

系统位	功能	描述
%SW101	模拟量输入	该字包含采集模拟量输入的值。其值的范围取决于 %SW100 的选择。 %SW100 = 0 %SW101 = 0 %SW100 = 1 %SW101 从 0 到 1000 变化 %SW100 = 2 或 4 %SW101 从 0 到 1000 变化 %SW100 = 3 或 5 %SW101 从 -10000 到 10000 变化
%SW102	模拟量输出	模拟量输出功能命令字。 值为 0: 正常 %PWM 操作 值为 1: 无量程操作 值为 2: 单极量程 模拟 %PWM 值为 3: 双极量程 该字必须由应用程序进行写操作。
%SW103	模拟量输出	该字包含将应用于模拟量输出的值。其值的范围取决于 %SW102 的设置。 %SW102 = 0 %SW103 = 0 %SW102 = 1 %SW103 在 5 和 249 之间 %SW102 = 2 %SW103 在 0 和 10000 之间 %SW102 = 3 %SW103 在 -10000 和 10000 之间 该字必须由应用程序进行写操作。
%SW110	加／减计数器	在输入 %I0.4 的上升沿读取计数器的值。
%SW111	高速计数器	第 1 位：1 = 高速计数器（直接）输出使能 第 2 位：1 = 选择频率计的时基（1 = 100ms，0 = 1s） 第 3 位：1 = 更新 %FC 频率（此位由用户复位到 0）
%SW114	调度模块使能	由用户程序或编程终端使能或禁止调度模块 (RTC) 的操作。 第 0 位：1 = 使能调度模块 #0 …… 第 15 位：1 = 使能调度模块 #15 初始时所有调度模块都是使能的。
%SW116	模拟量模块(EA4A2)设置	%SW116 的第 0～11 位对应模拟量模块的壮志位置： 15 0 位置 3 位置 2 位置 1 相应的位为 0 则模拟量输入为电压输入，相应的位为 1 则模拟量输入为电流输入

系统位	功能	描述
%SW117	模拟量模块(EAV8A2/EAP8)设置	%SW116 的高 8 位为模拟量模块的输出信号设定： 15 8 7 0 * * *=000 模拟量输出为 4mA 恒定电流输出 *=01 模拟量输出通道 0 为 0～2mA 可调，通道 1 为 4mA 恒定电流输出 *=02 模拟量输出通道 0,1 均为 0～2mA 可调 %SW117 的第 8 位模拟量模块的输入信号设定： 8 位分别对应于 8 路模拟量输入： 相应的位为 0 则模拟量输入为 0～5V 电压输入； 相应的位为 1 则模拟量输入为 PT100 温度信号输入
%SW118	主 PLC 状态	显示主 PLC 上检测到的故障。 第 0 位：0 = 其中一个输出断开 第 3 位：0 = 传感器电源故障 第 8 位：0 = Neza 内部故障或硬件故障 第 9 位：0 = 外部故障或通讯故障 第 11 位：0 = PLC 执行自检 第 13 位：0 = 配置故障（I/O 扩展已配置但不存在或错误） 这个字的其它所有位为 1 而且保留未用。因此对于一个没有故障的 PLC，这个字的值为:16#FFFF。
%SW119 态	对等 PLC I/O 的状	显示对等 PLC I/O 上检测到的故障（这个字只能被主 PLC 使用）。这个字各位的分配和 %SW118 相同，除了： • 第 13 位：没有意义 • 第 14 位：尽管对等 PLC 在初始化时还存在，现在丢失。(2)

6.编程指南

6.1 编程建议

处理程序的跳转

要注意避免使用程序跳转，循环将需要很长的时间并可能增加扫描时间。要避免程序向上跳转。

输出程序

输出位或内部位在程序中只能控制一次。当输出更新时，只有输出位的最后一次扫描值有效。

使用直接连接的紧急停止传感器

直接用于紧急停止的传感器不能由 PLC 来控制，它们必须直接连接到相应的控制回路。

处理恢复供电

使恢复供电处于手动操作状态，因为自动重新启动可能导致装置的意外操作（使用系统位 %S0、%S1 和 %S9）。

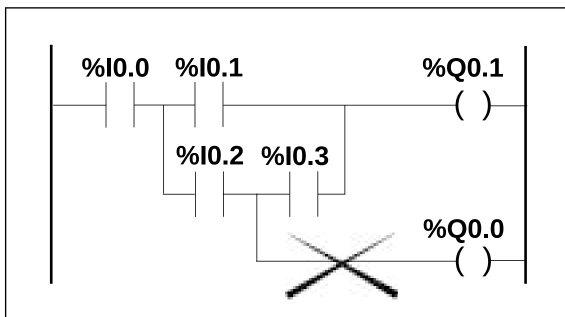
时间和调度模块的管理

%S51 位给出了任何调度模块的故障，因此应该对它进行检查。

注意：当一个程序输入以后，终端要检查它的指令语法，操作数以及指令和操作数之间的组合。终端诊断功能将检查编程错误（参见附录）。

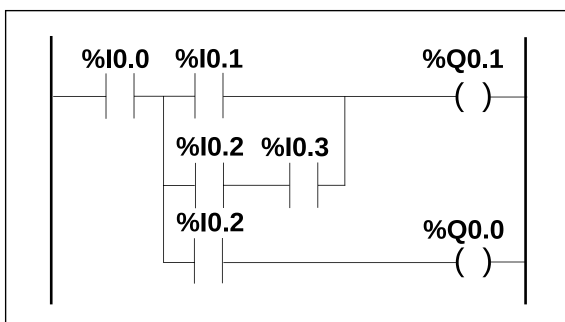
使用圆括号时的另外几点注意事项

• 赋值操作不能放在圆括号中。




```
LD      %I0.0
AND     %I0.1
OR(     %I0.2
ST      %Q0.0
AND     %I0.3
)
ST      %Q0.1
```

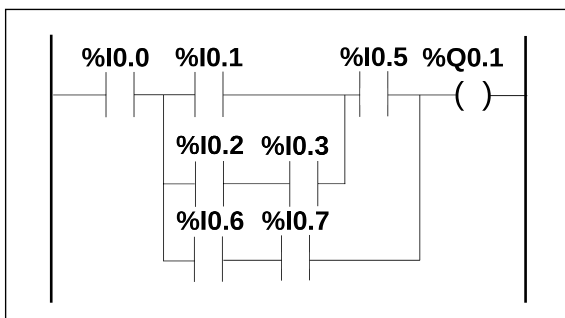
为了执行同样的功能，可以编写如下的等价程序：



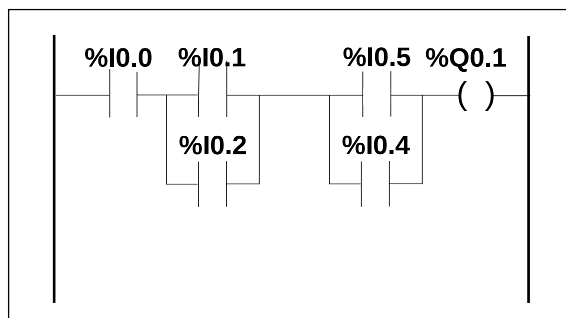

```
LD      %I0.0
MPS
AND(    %I0.1
OR(     %I0.2
AND     %I0.3
)
)
ST      %Q0.1
MPP
AND     %I0.2
ST      %Q0.0
```

• 如果好几个触点平行放置，它们要么相互嵌套，要么完全分离。

示例 1

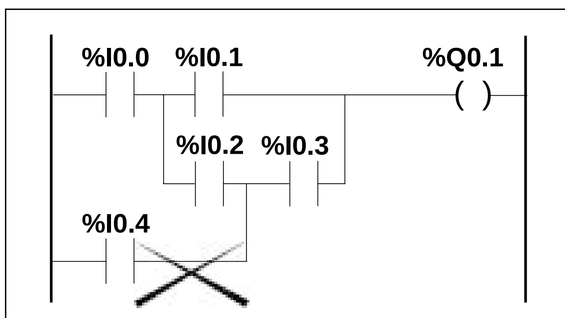


示例 2

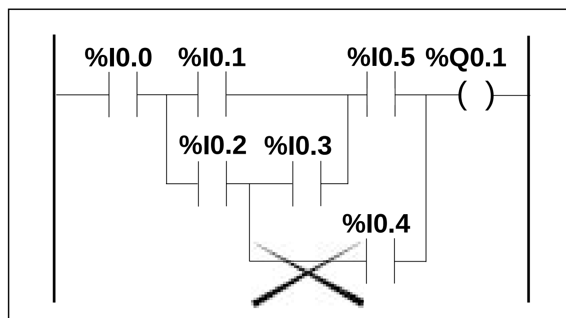


• 不能编写成下图的形式。

示例 3

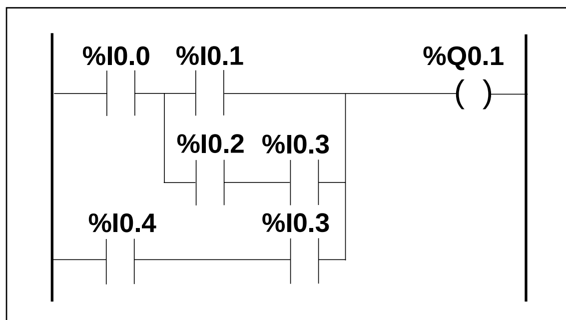


示例 4



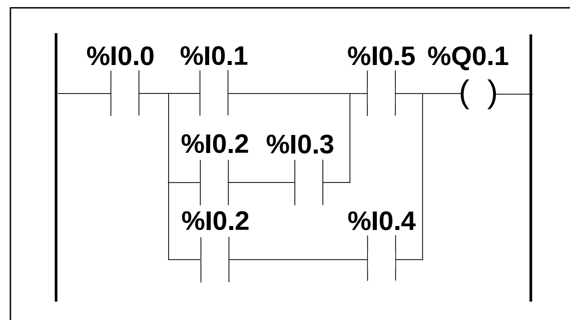
为了执行与上一页等价的图表，必须进行以下的修改：

示例5（参见示例3）




```
LD      %I0.0
AND(    %I0.1
OR(     %I0.2
AND     %I0.3
)
)
OR(     %I0.4
AND     %I0.3
)
ST      %Q0.1
```

示例6（参见示例4）

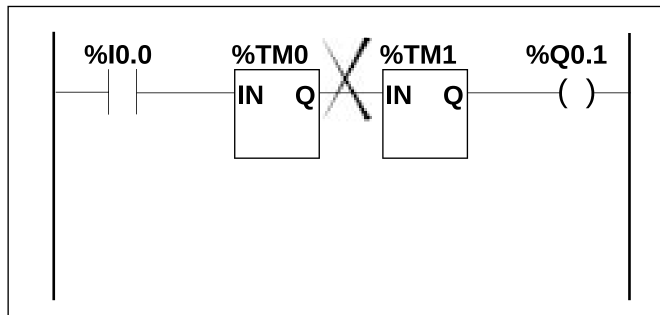



```
LD      %I0.0
AND(    %I0.1
OR(     %I0.2
AND     %I0.3
)
AND     %I0.5
OR(     %I0.2
AND     %I0.4
)
)
ST      %Q0.1
```

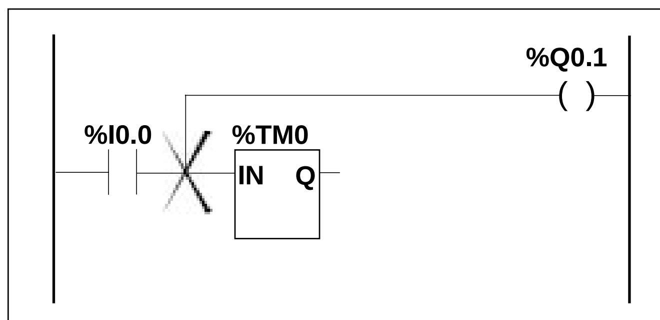
B

6.2 可逆性条件

检查如下条件，以使程序完全可逆 (1):



- 以下命令不可逆：
  XOR、XORN、XORF、XORR
  JIMPCN、ENDCN 或 N。
- 功能模块应该使用可逆的编程方法（参见第2.2-2节）：
- 功能模块不能串联或嵌套。
- 赋值指令禁止用在BLK和OUT_BLK指令间，或BLK和END_BLK指令间(如果没有编写OUT_BLK)。



禁止的编程方法：

```
BLK  %TM0
LD   %I0.0
ST   %Q0.1
IN
END_BLK
```

允许的编程方法：

```
BLK  %TM0
LD   %I0.0
IN
END_BLK
LD   %I0.0
ST   %Q0.1
```

(1)注意：当一个指令列表程序不可逆时，它将保留在指令列表语言中，而余下的可逆程序将转化为梯形图。

6.3 可逆性规则

- 一个规范的梯级高度不能超过7个单元，宽度不能超过11个单元(7X11网络)
- 一个以LD开头的语句必须以条件动作指令结束。
- JMPCN、ENDCN、NOP和N指令是不可逆的。
- 圆括号内的动作指令是不可逆的。
- 圆括号内的栈指令是不可逆的。
- 动作指令后面的OR指令是不可逆的。
- RET、JMP、END是无条件指令，梯级中不能再有其它相同指令。
- 功能模块的输入、输出只能通过标准的可逆功能模块指令存取，功能模块的直接存取指令是不可逆的。
- END_BLK后面的顺序指令将导致该顺序不可逆。
- 使用AND和 OR指令的功能模块输出是不可逆的。
- 功能模块输出梯级中的OR指令必须放在圆括号中。

- BLK 和 END_BLK 之间的无条件动作指令是不可逆的。
- OUT_BLK 必须后接一个有效的功能模块输出的 LD 指令，或者是 END_BLK 指令。
- 在 MRD 或 MPP 指令之前要么是一个条件动作指令，要么是一个栈操作指令。
- 在 MPS 和 MPP 之间的非嵌套 OR 指令是不可逆的。
- 在 MPS 、MRD 或 MPP 指令后面的 OR 指令是不可逆的。
- 一个子程序调用或 JMPC 指令必须是一个梯级的最后动作指令。
- 当在一个指令列表前加上标题或注释时，只能是一个标题行和最多四个注释行。如果一个空行放在标题行和指令列表之间的任何位置，则部分标题就可能不在对应的梯形图梯级注释区显示出来。
- 如果指令列表前面的注释行多于四行，则第五个注释行将被认为是一个标题行，并且上面的注释行将不显示在对应的梯形图梯级的注释区。

当一个指令序列不可逆时，它仍然保留在指令列表语言中，而剩下的可逆程序转换为相应的梯形图，如下图所示。



7.通信

7.1 TER 端口通信

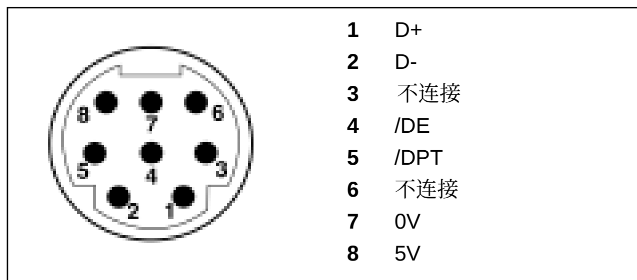
7.1-1 TER 端口特性

TSX NEZA 本体 PLC 的面板上有一个 TER 端口

- 端口特性

链路类型	: RS485
协议	: UNI-TE 格式 V2
波特率	: 9600/19200 bits/s 1 个起始位，8 个数据位，奇校验，1 个停止位
连接器类型	: 8 针快速连接 mini-DIN
最大链路距离：	
TSX08PALM	: 10 米
UNITELWAY 设备	: 10 米（无分线盒）
ASCII 设备	: 10 米

- 连接器引出线—— UNI-TELWAY 设备



/DPT信号 用来选终端端口的操作模式：

/DPT = 1　UNI-TELWAY Master 模式。

/DPT = 0　UNI-TELWAY Slave 模式 /ASCII 模式 /Modbus Master/Modbus Slave 模式

连接针 5 到 7，置 /DPT=0。

注意：

连接设备时，应先切断电源（除了编程终端）。

Modbus,ASCII 和 UNI-TELWAY 设备是专用的。

Modbus,ASCII 设备必须断开，才能使用编程器(PPC 或 PC)。

7.1-2 TER 端口的通用信息

Neza TER端口有五种不同操作模式：

- UNI-TELWAY Master
- UNI-TELWAY Slave
- ASCII
- MODBUS Master
- MODBUS Slave

TER 端口的操作模式在软件配置中选择，利用 Mini DIN 槽口上的信号 /DPT（针 n°5）：

- 当信号 /DPT 在 1（针 n°5 未连），TER 端口是在 UNI-TELWAY 主模式。
- 当信号 /DPT 在 0（针 n°5 连到针 n°7=0V），TER端口在 ASCII 模式或 UNI-TELWAY 从模式或者 Modbus 主／从模式。用编程工具 PC 和掌上电脑编程器 PPC 由软件配置进行选择（默认值是 Modbus Slave 模式）。

/DPT 状态以系统位 %S100 表示。

配置编程端口的对话框如下：

编程端口

协议类型 (T)
ASCII　UNI-TELWAY Master　UNI-TELWAY Slave
Modbus Slave　Modbus Master

波特率 (B)
1200　2400　4800　9600　19200

Modbus/UNI-TELWAY Slave 基地址: 4

校验 (P)
奇
偶
无

数据位 (D)
8 位　7 位

停止位 (S)
1 位　2 位

UNI-TELWAY 超时
Master 和 Slave 方式 (字符): 30

确定 (O)　取消 (C)

7.2 TSX Neza 在 ASCII 模式下

在 ASCII 模式下，TSX Neza 可以与一些设备（打印机或终端）进行一些简单的通信，发送和／或接收字符串。

> 该模式进行点到点连接操作。

TER 端口配置可以在 PLC 配置对话框中修改：

- 类型：半双工
- 速率： 1200, 2400, 4800, 9600 或 **19200** 波特
- 格式：1 个起始位，7 或 **8** 个数据位，**1** 或 2 个停止位
- 校验：偶校验，奇校验，无校验

上面的粗体值是默认值。

ASCII 配置对话框：

编程端口

协议类型 (T)
ASCII
UNI-TELWAY Master
UNI-TELWAY Slave
Modbus Slave
Modbus Master

波特率 (B)
1200
2400
4800
9600
19200

Modbus/UNI-TELWAY Slave 基地址: 4

校验 (P)
奇
偶
无

数据位 (D)
8 位
7 位

停止位 (S)
1 位
2 位

UNI-TELWAY 超时
Master 和 Slave 方式 (字符): 30

确定 (O)
取消 (C)

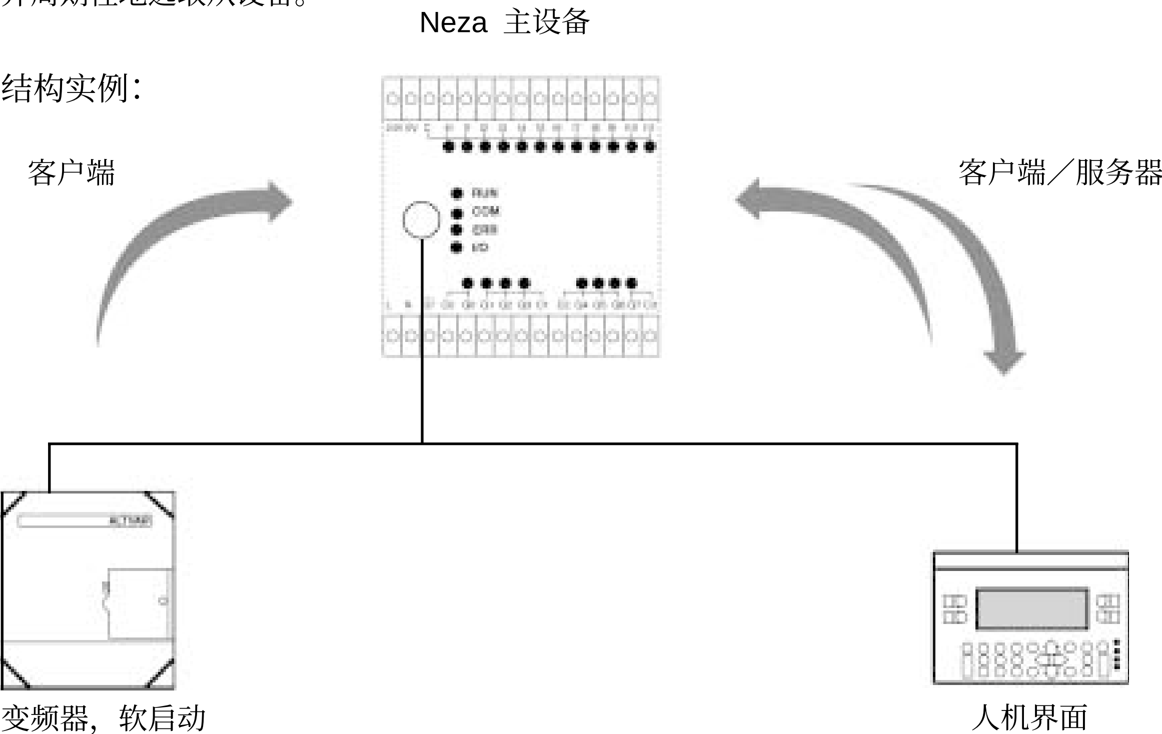
在 ASCII 模式下编程可以使用 EXCH 指令，就象使用功能块 %MSG 一样。

7.3 TSX Neza 在 UNI-TELWAY模式下

7.3-1 TSX Neza Master 在 UNI-TELWAY 模式下

TSX Neza 与其他的 UNI-TELWAY 设备是兼容的，如人机接口(MMI)，识别设备和驱动器。该部分提供用户在 UNI-TELWAY 总线上连接这些设备的通用指导。详细的资料，参见相关资料。当 TSX Neza 是连接到 TSX Neza 终端端口从设备的 UNITELWAY 通信主设备时，将控制网络并周期性地选取从设备。

结构实例：



通常 TSX Neza 是 UNI-TE 服务器。但是也可以提供有限的 UNI-TE 客户端服务。

- 配置

多数 UNI-TELWAY 设备需要两个地址。第一个地址是用户用开关物理设置的。另一个地址通常是物理地址加 1。TSX Neza 可以与编程终端通信，最多可有两个附加 UNI-TELWAY 设备。

- 地址分配 ：

0 : TSX08 Neza （通信主设备）
1 : 编程工具(TSX08PALM 掌上电脑编程器或 PC 电脑（带 **PL707WIN** 软件））
2-3-5 : 客户设备
4 : 客户和／或服务器设备
2-3-4-5 : 客户和／或服务器设备

可以连接到不同地址的设备概述

参考	允许地址		
设备	**1**	**2,3**	**4,5**
编程终端	Yes	No	No
MMI	No	Yes	Yes
识别设备	No	Yes	Yes
速度控制器	No	Yes	Yes

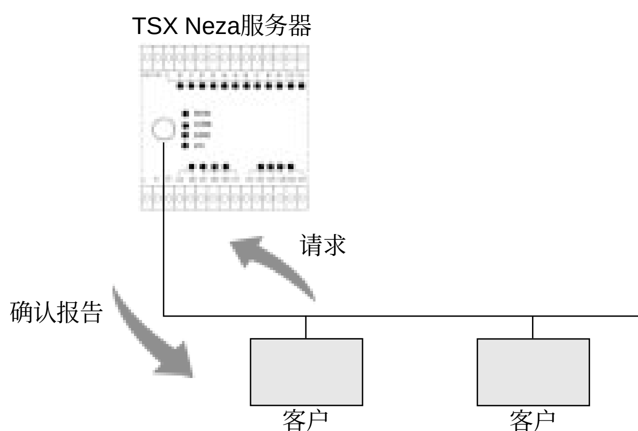
该 TER 端口配置不能在 UNI-TELWAY Master 模式下进行更改。它根据下面几种传送模式管理 5 个从链路地址：

- 类型：半双工
- 速率：9600 波特
- 格式：1 个起始位，8 个数据位，1 个停止位
- 校验：奇校验

  超时参数可在配置屏幕中设置。

- **UNI-TE** 服务器功能：

TSX Neza 将响应客户端发出的命令。客户端是可以与 TSX Neza 通信的智能设备，可以从 TSX Neza 写数据或读数据。



寻址

TSX Neza 地址总是为 0 （主）

- TSX Neza 轮询地址 1 到地址 5。
- 编程工具（ PPC 或 PL707WIN 软件)地址总是 1。
- 通常，UNI-TELWAY 设备是由分线盒中的开关或缆线连接设置。

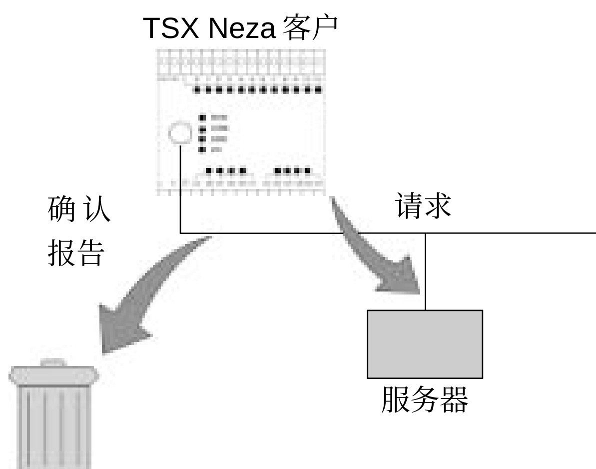
注意：

- 当总线上的主设备是 TSX Neza 时，从设备间不能进行通信。

所有的总线设备可以通过目标地址 0.254.0 询问TSX Neza 服务器。TSX Neza UNITE 服务器仅响应发往该地址的帧。其他目标地址的消息将被忽略。

• **TSX Neza** 作为客户：

TSX Neza 可以与服务器进行通信。服务器是智能化设备，将执行 TSX Neza 发送的命令。



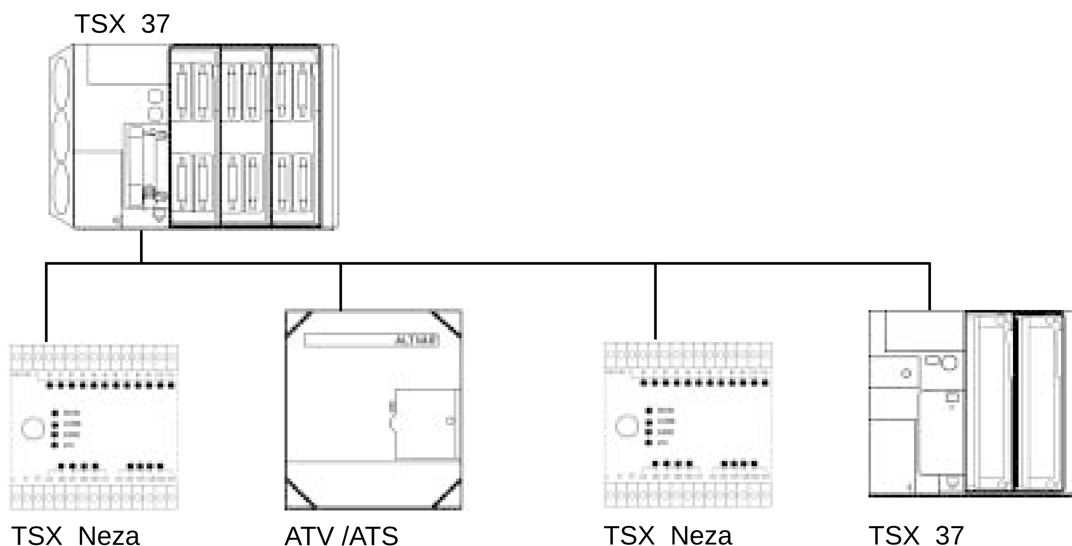
TSX Neza 主设备可以用 EXCH 指令向任何（地址在 1 和 5 之间）的从设备发送请求。它的源地址是 0.254.16 。

可以利用 EXCH 指令，与信息交换控制块 %MSG 一样在 UNI-TELWAY 主模式下编程。

7.3-2 TSX Neza Slave 在 UNI-TELWAY模式下

它用在几个设备同时, 多点连接中(如 PLC ， 编程终端， MMI 设备， 速度控制器等)。

结构实例



与主模式不同，在UNI-TELWAY从模式下，终端端口配置可以在PLC的配置对话框中，由用户用编程工具 PL7 07 WIN 或 TSX08PALM进行更改：

- 类型：半双工
- 速率：1200 ，2400 ，4800 ，9600 或 **19200 bds**
- 格式：**1** 个起始位，7 或 **8** 个数据位，**1** 或 2 个停止位
- 超时：**30** 至 255 （见 1.6 ）
- 地址：**4** （ 1 到 97 ）
- 校验：偶校验，奇校验，或无校验

以上，粗体字为默认值

TSX Neza 使用 2 个连续逻辑地址：

- AD0 ： 基地址（配置地址），作为服务器地址。任何（本地或远端）设备都可以用该地址作为目标地址寻址 TSX Neza 系统服务器。
- AD1 = AD0 + 1 也称为客户地址。用该地址， TSX Neza 可以用 EXCH指令将请求发送到任何 UNI-TELWAY 总线设备（主设备或从设备）。



在配置屏幕中作的终端端口特性的更改，在电源重启或在 Mini DIN 插座上的 /DPT 信号修改发生改变时（终端端口断开连接／重新连接）才发生作用。

7.3-3 UNI-TELWAY 超时

UNI-TELWAY 链路层协议（主或从）使用超时。超时响应：

- 默认值为 30 个字符。当 TSX Neza 没有应用程序时，该周期增加到 125 个字符。

一旦发送了一帧，设备（主或从）就开始计时。如果在超时结束前没有收到应答，交换就没有被确认。如果这样的话，发送方将重发这一帧。

端口编程对话框用来设定30 到 8000 字符间的超时。

30 与 250 之间的值与等价数量的字符对应。251 与 255 之间的值与下面的值对应：

- 251 = 500 字符
- 252 = 1000 字符
- 253 = 2000 字符
- 254 = 4000 字符
- 255 = 8000 字符

在开电源或在修改 /DPT 信号时，配置值载入到系统字 %SW14 低位字节（忽略高位字节）。这样，超时值可以通过在系统字 %SW14 中写入新值来调整。在 PLC 扫描结束时起作用。

较长的超时值使 TSX Neza 可连接较慢的设备，如调制解调器类型的设备。

用系统字 %SW14 可以将超时的值降低到 10 个字符。

要点:

通过 UNI-TELWAY 连接某些设备（如调制解调器）到 TSX Neza 时，它们的响应时间决定超时参数必须设置在 255（ 9600 bd/s 时，约为 8 秒）。

该值防止 TSX Neza 和编程终端用 UNI-TELWAY 链路标准参数进行并发通信。

因此，很有必要：

- 在 Windows/95/98/2000 中的 DUNTLW.001 文件或用 Windows NT 中 XWAY 工具修改 AUTOSPEED 参数值（ 0 代替 1），和 UNI-TELWAY 的地址数（ 5 而不是 3 ）。
- 修改 PLC UNI-TELWAY 链路的超时值
- 在 Windows /95/98/2000 中的DUNTLW.001或用 Windows NT 中 XWAY 工具重新分配初始值。

7.3-5 TSX Neza（服务器）支持的 UNI-TE 请求

标准请求

服务	请求	请求		确证		含义
		Hex.(1)	Dec.(2)	Hex.	Dec.	
数据（读）	读 1 个位	00	00	30	48	读 1 个 %Mi 位
	读 1 个字	04	04	34	52	读 1 个 %MWi 字
	读对象	36	54	66	102	读对象(%Mi, %Mi:L, %Mwi, %Mwi:L)
数据（写）	写 1 个位	10	16	FE	254	写 1 个 %Mi 位
	写 1 个字	14	20	FE	254	写 1 个 %MWi 字
	写对象	37	55	FE	254	写对象 (%Mi, %Mi:L, %Mwi, %Mwi:L)
操作模式	运行	24	36	FE	254	设置设备运行
	停止	25	37	FE	254	设置设备中止

特殊请求

服务	请求	请求		确证		含义
		Hex.	Dec.	Hex.	Dec.	
数据（读）	读 1 个系统位	01	01	31	49	读 1 个 %Si 位
	读 1 个常量字	05	05	35	53	读 1 个 %KWi 字
	读 1 个系统字	06	06	36	54	读 1 个 %SWi 系统字
数据（写）	写 1 个系统位	11	17	FE	254	写 1 个 %Si 位
	写 1 个系统字	15	21	FE	254	写 1 个 %SWi 字

(1)Hex.十六进制　　(2)Dec.十进制

7.4 TSX Neza 在 MODBUS 模式下

7.4-1 MODBUS/JBUS 特性

MODBUS 链路特性：

- 物理层： RS485 无隔离，最大长度 200 米
- 链路层： 异步传输
  ASCII 帧（7 位）或 **RTU**（**8** 位）
  数据速率：1200，2400，4800，**9600** 或 19200 bds
  校验：偶，奇或无校验
  停止位：**1** 或 2
  字符间隔时间：**3**（1 到 127）字符
- 物理配置： 最多 32 个设备
  247 个逻辑地址（**1** 到 247）
- 服务： 位：128 位
  字：120 字（16 位）
  安全：由控制参数（CRC16）对每帧（在 RTU）进行校验

以上粗体值为默认值。

注意：设备连接时必须断开电源。

概述

对 MODBUS 和 JBUS 设备，允许 MODBUS 和 JBUS 通信，具有数据交换服务(%Mi 和 %MWi)。

MODBUS / JBUS 协议允许主设备和从设备间的数据交换，但不允许从设备间的直接通信，也不允许网络通信。

在 **ASC II** 模式下 **MODBUS** 通信

- ASC II 模式可以连接一些简单设备（如显示器等），其帧长是在 RTU 模式下帧长的 2 倍。

ASC II 帧的结构

头代码(3A)	从设备号	功能码	数据	LRC	CR	IF *
1 字节	2 字节	2 字节	2n 字节	2 字节	1 字节	1 字节

* 系统字 %SW67 用来设置结束分隔符参数(LF)。系统冷启动时，写入 16#000A。
只要Modbus Master 使用的不是16#000A的分隔符，用户就可以通过编程或调整来修改这个系统字。

在**RTU** 模式下的 **MODBUS** 通信 :

- 在 RTU 模式下的帧没有头代码字节和信息结束分隔符

从设备号	功能码	数据	CRC16
1 字节	1 字节	n 字节	2 字节

B

7.4-2 TSX Neza Slave 在 MODBUS 模式下

7.4.2-1 配置 MODBUS Slave

从 PL7 07 WIN 的“**配置**”菜单中，选“**编程端口**”配置 MODBUS / JBUS 的连接，出现下面的对话框：

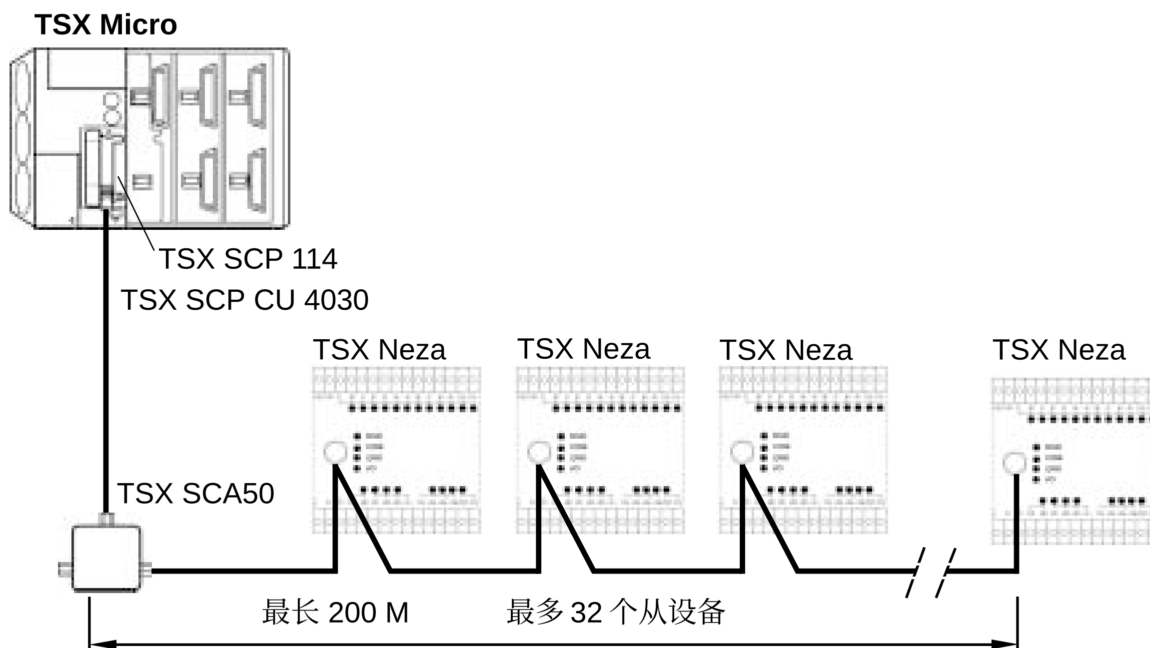


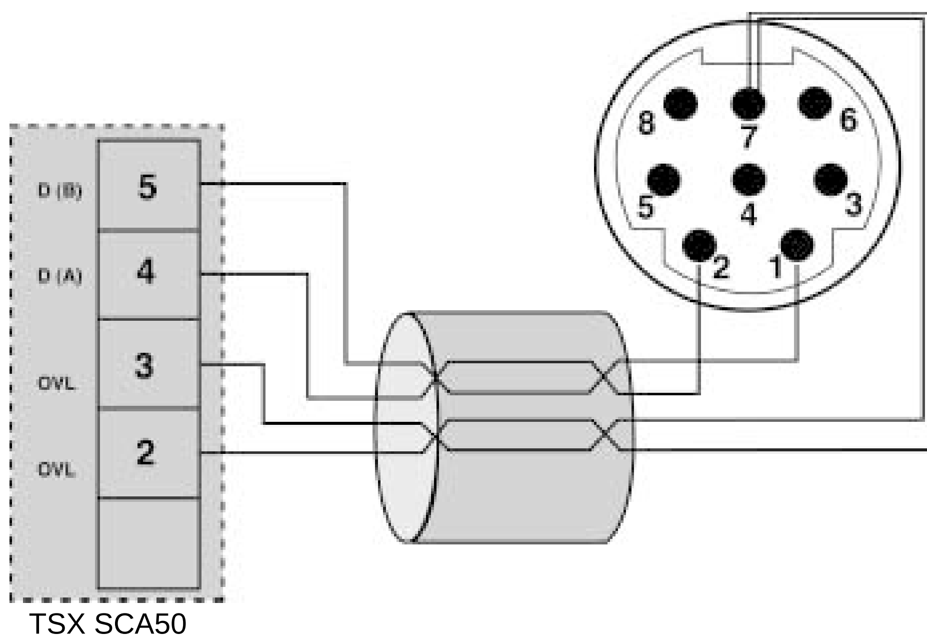
选中“ Modbus Slave ”并配置下面的参数。

在动态操作中，不能修改参数（速率和格式）。从设备的速率不能根据主设备的速率自动调整。任何速率和协议格式的修改在配置菜单启动时立即生效。

7.4.2-2 MODBUS Slave 连接

举例　　菊花式连接





7.4.2-3 MODBUS Slave 中，TSX Neza 支持的请求

TSX Neza 支持下面列表中请求，其于的将以异常编码 01 :（功能不可知）加以拒绝。

MODBUS 功能一次仅仅处理一个请求，如同在 MODBUS 协议中一样，主设备必须等待从设备响应或超时方可启动向从设备发送新信息。

标准 **MODBUS** 请求：

- 01/02 : 读 n 个连续内部位：访问位 %M0 到 %M127
- 05 : 写 1 个内部位：访问位 %M0 到 %M127
- 15 : 写 n 个连续内部位：访问位 %M0 到 %M127
- 03/04 : 读 n 个连续内部字：访问字 %MW0 到 %MW255
- 06 : 写 1 个内部字：访问字 %MW0 到 %MW255
- 16 : 写 n 个连续字：访问字 %MW0 到 %MW255

请求的具体说明见 2.3。

TSX Neza 到 UNITE 服务器的访问请求：

- 0F: 识别
- 4F: 读 CPU
- 24: 运行
- 25: 停止
- 33: 初始化

异常代码：

当从设备不知道怎样处理请求时，就会反回异常代码。

应答帧包含：

- 接受的功能码，加上 16#0080
- 错误类型的异常代码

TSX Neza处理的两种异常代码：

- 01:功能未知(TSX Neza 不支持的请求）
- 03:无效数据（位或字的数据错误，如写位时数据既非 16#FF00，又非 16#0000）

7.4.2-4 COM LED 状态

一旦发出的请求有了应答，TSX Neza 的 COM LED 点亮 50 毫秒。

点亮频率由主设备和从设备之间的数据交换情况决定。

7.4.2-5 相关的系统位和系统字

当处理 MODBUS 请求时，TSX Neza 将系统位 %S70 设为 1。

用户可以使用该位，将该位复位为 0 需由用户操作。

系统字 %SW67 可用来在 ASCII 模式下设置结束分隔符参数(LF)。

在系统冷启动时，写入 16#000A。

用户可以在主设备的结束分隔符不是 16#000A 时，通过编程或调整来修改这个系统字。

7.4-3 TSX Neza Slave 到 UNI-TE 服务器的请求

该规则包含了在 MODBUS 请求中编码 UNITE 请求，以查询 UNITE 服务器，该访问平时是由 TER 端口支持的。

使用的是特殊功能代码：**65**

规则：

MODBUS 主设备到从设备的请求

功能号： 65	请求码	类别码	数据

MODBUS 从设备对主设备的应答

功能号： 65	应答码	应答数据

类别码总是为 7 。

UNITE 应答码可用来查找服务器操作的结果。

有三种情况：

- 请求应答码 = 请求码 +30H：操作已由服务器执行，附加信息给出了操作结果的详细信息。
- 正常应答码 =16#00FE :操作已经正确执行，应答中没有附加信息。
- 异常应答码 =16#00FD :服务器没能执行操作（请求不可知，超出数值范围等）。

7.4.3-1 识别请求

识别请求用来向 UNITE 服务器发出请求的设备，提供识别和结构信息。通过 LED 状态和设备模块状态的显示，识别请求还提供基本的诊断信息。

请求码：**0F**

应答码：**3F**

报告格式（27 字节）:

- 识别类别（字节）：字节总是等于 H‘FF’。
- 产品范围（字节）：该参数指明设备属于哪一类产品：H‘0B’=TSX Neza 范围。
- 版本（字节）：设备版本号，以两个 BCD 4-bit 字节编码。
- **ASCII** 串（字符串）：指明产品参考。

在 TSXNeza 上，它包含一个字节(H‘10’)，后面跟着的是 ASCII 文本（16 字节）：
‘TSX 081’
‘1-’ 有实时时钟调度模块，‘0’ 没有实时时钟调度模块。
‘20’ 20I/O
H‘20’
H‘20’

- 设备状态(字节)：对于 TSX Neza，该字节总是等于 0。
- **LED** 状态：该字节指明 TSX Neza 的 4 个状态指示 LED 的状态。

RUN（位 0 和 1），ERR（位 2 和 3），I/O（位 4 和 5），COM（位 6 和 7）。
每一个 LED 是以两位编码。
00 : 灭
01 : 闪烁
10 : 亮
11 : 无意义

- 基本模块类型（字节）：H‘30’（类型 = CPU）
- 制造商参考（字节）：H‘0A’
- 类别参考（字节）：定义设备类型：

H’50’ 20 I/O 不带实时时钟
H’51’ 20 I/O 带实时时钟

- 基本模块状态(8 位表)：该表中的值符合配置设备的标准。

置于 1 的位表明有错误。

位号	类别	注释
0	ERR-INT	1: TSX Neza 硬件错误
1	ERR-EXT	1:传感器电源错误
2	保留	
3	MOD-At	1:模块自检
4	保留	
5	ERR-CNF	1:已配置的扩展不存在或错误
6	保留	
7	保留	

- 连接设备数(字节)：由于没有子设备，所以该值等于 0。

7.4.3-2 读 CPU

该功能用来诊断 TSX Neza PLC 处理器的状态。

请求码：**4F**
应答码：**7F**

报告格式（14 字节）：
- 扩展（字节）：用来传输通信传送编号：无意义
- **LED** 状态（8 位表）：见上文识别请求中的 LED 状态
- **PLC** 状态（8 位表）：描述 PLC 的物理状态

位 0 :运行(1)，停止(0)
位 1 : 应用程序可执行(1)
位 2 :有存储卡（总是为 1）
位 3 :有强制操作(1)
位 4 :保留：0
位 5 :软件错误(1)
位 6 :保留：0
位 7 :保留：0
- 保留地址（6 字节表）：描述保留给处理器的应用实体的网络地址。对于 TSX Neza，每字节值为 16#00FF，表示处理器没有保留。
- 应用程序错误类型（字节）：总是为 0。
- 调试信息（8 位表）：

位 0 :强制状态（1: 有强制，0: 无强制）
位 1 到 7: 总是为 0
- 产品范围（字节）：该参数指示设备所属的产品范围：H‘0B’=TSX Neza 范围。
- 应用程序 **/ PLC** 信息(8 位表)：

位 0 :应用程序在 RAM 中(1)
位 1 :程序在 RAM 中，检查和 OK(1)
位 2 :程序在 RAM 中，可执行(1)
位 3 :程序在 RAM 中，受保护(1)
位 4 :程序在 RAM 中，与 FLASH 中程序不同(1)

位 5 :应用程序与 OS 兼容 (1)
位 6 :有实时时钟调度模块(1)
位 7 : 0
• 备份信息(8 位表):
位 0 : 应用程序存在(1)
位 1 : 检查和 OK (1)
位 2 : FLASH 应用程序可执行(1)
位 3 : FLASH 应用程序受保护(1)
位 4 : 自动载入应用程序(1)
位 5 : 应用程序与 OS 兼容(1)
位 6 : 0
位 7 : 0

7.4.3-3 RUN 请求

请求码: **24**
应答码: **FE**
如果发生以下情况会出现异常结果:
• RUN/STOP 输入配置在 STOP
• 应用程序无法执行
• 软件错误(警戒时钟溢出,调用不存在的 G7 步骤)。
但发送该请求至一个已经运行的 TSX Neza 不算是错误。

7.4.3-4 STOP 请求

请求码: **25**
应答码: **FE**
如果在 RAM 中的应用程序不能执行,会得到异常结果。
在一个已经停止的 TSX Neza 中发 STOP 指令不算错误。

7.4.3-5 INIT 请求

请求码: **33**
应答码: **63**
• 初始化类型(发送字节):必须总是为 1。
• 报告: 指出初始化操作的结果:
00 :初始化 OK
01 :初始化类型值不是 1

7.4-4 TSX Neza Master 在 MODBUS 模式中

7.4.4-1 配置 MODBUS Master

从 **PL707WIN** 的 “ 配置 ” 菜单中选 “ 编程端口 ” 配置 MODBUS/JBUS 的连接，出现下面的对话框

编程端口

协议类型 (T)
ASCII　UNI-TELWAY Master　UNI-TELWAY Slave
Modbus Slave　Modbus Master

波特率 (B)
1200　2400　4800　9600　19200

Modbus/UNI-TELWAY Slave 基地址: 4

校验 (P)
奇
偶
无

数据位 (D)
8 位　7 位

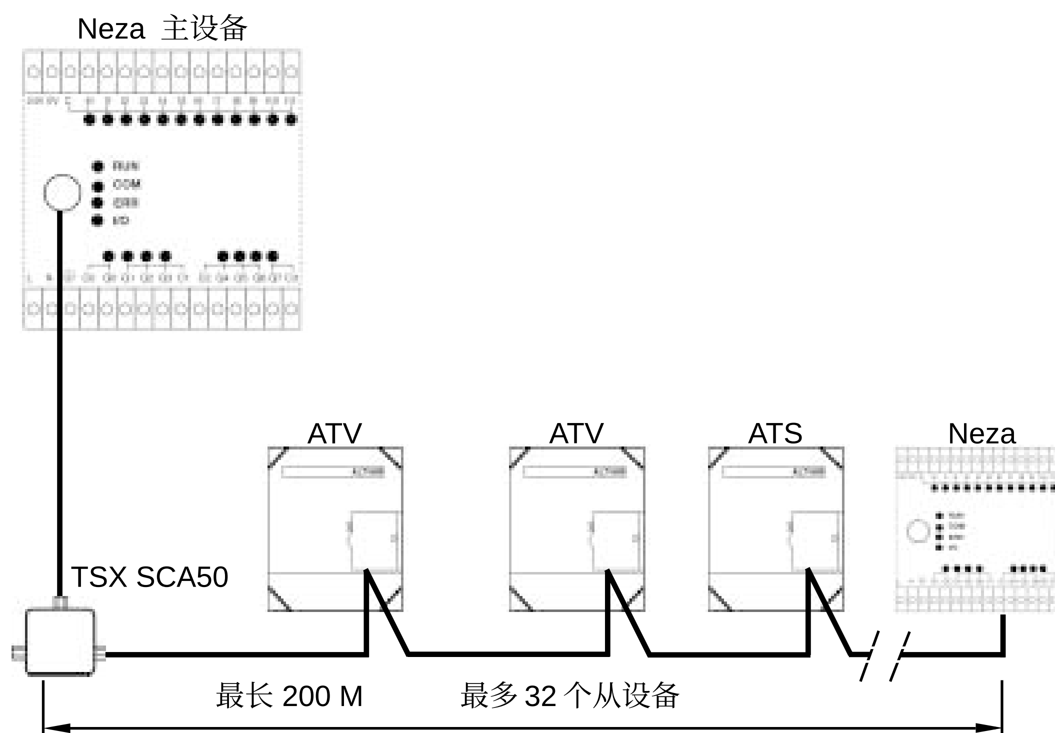
停止位 (S)
1 位　2 位

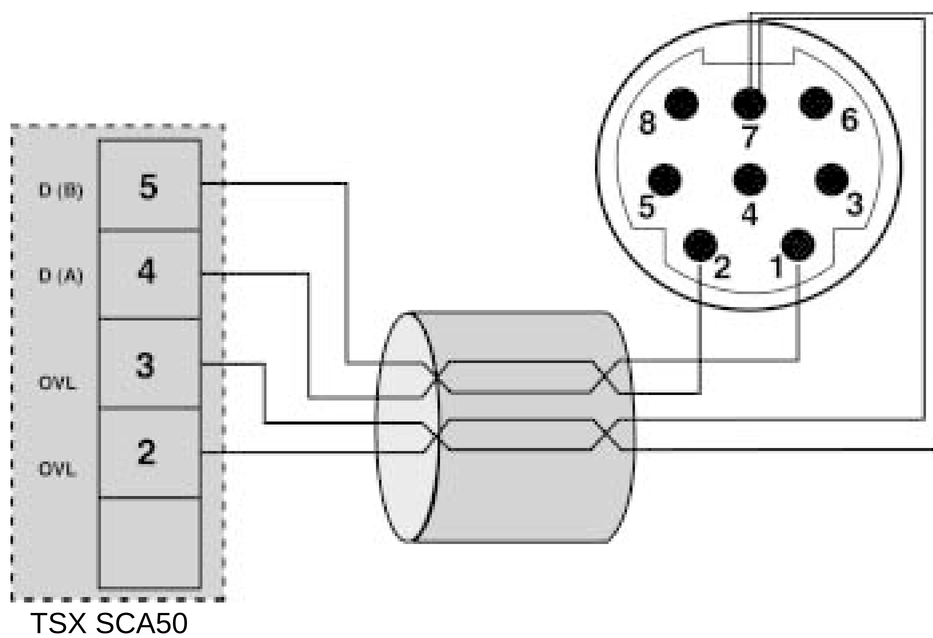
UNI-TELWAY 超时
Master 和 Slave 方式 (字符): 30

确定 (O)　取消 (C)

选中 “ Modbus Master”，配置下面的参数。
在动态操作中，不能修改参数（速率和格式）。从设备的速率不能根据主设备的速率自动调整。
任何速率和协议格式的修改在配置菜单启动时立即生效。

7.4.4-2 MODBUS Master 连接





7.4.4-3 MODBUS Master 中，TSX Neza 支持的请求

TSXNeza 使用 %MSG 和 %EXCH 指令支持 Modbus Master 功能。

%MSG 的使用方法与 ASCII 和 UNI-TELWAT 通信中使用方法一样。

%EXCH 是在处理 Modbus 报文时使用的。只有 %MW 可用于控制Modbus Master。(Modbus Master 通信时，必须使 DTP=0)

举例：[EXCH%MWn:L]

控制和接收字表的定义：

<table>
<tr><th></th><th>高位字节</th><th>低位字节</th></tr>
<tr><td>%MWn</td><td>从地址</td><td>Modbus 功能代码</td></tr>
<tr><td>%MWn+1</td><td colspan="2">从 PLC 的起始寄存器（从 0 到 65535，0 代表 Modicon 中的 1 或 TE 中的 0）</td></tr>
<tr><td>%MWn+2</td><td colspan="2">数据长度（字：1~125，位：1-2000）</td></tr>
<tr><td>%MWn+3</td><td>重试次数(0~3)</td><td>在从设备应答前超时</td></tr>
<tr><td>%MWn+4</td><td rowspan="5">数据区
• 准备写入从设备的数据
• 由从设备读出的数据</td><td>字 0 或位 0~15</td></tr>
<tr><td>%MWn+5</td><td>字 1 或位 16~31</td></tr>
<tr><td>%MWn+6</td><td>字 2 或位 32~47</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>...</td><td>...</td></tr>
</table>

Modbus Master 功能代码：

- 01 读线圈状态
- 02 读输入触点状态
- 03 读保持寄存器
- 04 读输入寄存器
- 05 强置 1 个线圈
- 06 预设 1 个寄存器
- 15 强置多个线圈
- 16 预设多个保持寄存器

... ...

L 必须大于 4+ 数据区，否则在状态字中将显示错误代码。

如果 Modbus 从地址 = 0，表示是广播式报文。MSG.D 在报文发送出去后置 1。主设备不需要接受应答报文。

超时：1~255 代表 10ms-2.55s，如果该值为 0,则代表 10s，该值在 %MWn+3 的低位字节中设置。

7.4.4-4 COM LED 状态

一旦发出的请求有了应答，TSX Neza 的 COM LED 点完 50 毫秒，点亮频率由主设备和从设备之间的数据字交换情况决定。

7.4.4-5 相关的系统字

系统字 %SW67 可用来在 ASCII 模式下设置结束分隔符参数(LF)。

在系统冷启动时，写入 16#000A。

用户可以在主设备的结束分隔符不是 16#000A 时，通过编程或调整来修改这个系统字。

7.4-5 标准 MODBUS 请求

这些请求可用来通过访问 TSX Neza 的对象 %Mi 和 %MWi 来交换 MODBUS 对象。

在 RTU 模式下的这些请求在下面具体叙述。在 ASCII 模式下，数据相同，CRC16 被 LRC 所代替。

7.4.5-1 读 n 个内部位 %Mi

功能 **01** 或 **02**

请求：

从设备号	1 或 2	第 1 个位的号码		位的个数		CRC 16
		PF	Pf	PF	Pf	
1 字节	1 字节	2 字节		2 字节		2 字节

应答：

从设备号	1 或 2	读的字节数	值	●●●●●●●●●●●●	值	CRC 16
1字节	1字节	2字节				2 字节

实例：从 2 号从设备中读位 %M3 的值

请求	02	01	0003	0001	CRC 16
应答	02	01	01	xx	CRC 16

xx：
- 00如果%M3 = 0
- 01如果%M3 = 1

7.4.5-2 读 n 个内部字 %MWi

功能 03 或 04

请求:

从设备号	3 或 4	第一字的号码		字的个数		CRC 16
		PF	Pf	PF	Pf	
1 字节	1 字节	2 字节		2 字节		2 字节

应答:

从设备号	3 或 4	读的字节数	第一个字的值		•••••••••••••	最后一个字的值		CRC 16
			PF	Pf		PF	Pf	
1 字节	1 字节	1 字节	2 字节			2 字节		2 字节

实例：从 6 号从设备中读字 %MW20 到 %MW24 的值。

请求	06	04	14	05	CRC 16

应答	06	04	0A	xxxx	•••••••••••••	xxxx	CRC 16
				%MW20 的值		% MW24 的值	

7.4.5-3 写 1 个内部位 %Mi

功能 **05**

请求：

从设备号	5	位的号码		位的值＊	CRC 16
		PF	Pf		
1 字节	1 字节	2 字节		2 字节	2 字节

＊“ 位的值 ” 栏内有两个可能值：

- 该位为 0 = 0000，
- 该位为 1 = FF00.

应答：

从设备号	5	位的号码		位的值	CRC 16
		PF	Pf		
1 字节	1 字节	2 字节		2 字节	2 字节

实例：在 2 号从设备的位 %M3 中写入值 1

请求	02	05	03	FF00	CRC 16
应答	02	05	03	FF00	CRC 16

7.4.5-4 写 1 个内部字 %MWi

功能 06

请求:

从设备号	6	字的号码		字的值		CRC 16
		PF	Pf	PF	Pf	
1 字节	1 字节	2 字节		2 字节		2 字节

应答:

从设备号	6	字的号码		字的值		CRC 16
		PF	Pf	PF	Pf	
1 字节	1 字节	2 字节		2 字节		2 字节

实例：在 5 号从设备中的字 %MW12 中写入值 16#3A15

请求	05	06	0C	3A15	CRC 16
应答	05	06	0C	3A15	CRC 16

7.4.5-5 写 n 个内部位 %Mi

功能 15

请求:

从设备号	0F	起始位地址	位的个数	字节数	位的值	CRC 16
1 字节	1 字节	2 字节	2 字节	1 字节	n 字节	2 字节

应答:

从设备号	0F	起始位地址	位的个数	CRC 16
1 字节	1 字节	2 字节	2 字节	2 字节

7.4.5-6 写 n 个内部字 %MWi

功能 **16 (H'10')**

请求:

从设备号	10	第 1 个字的号码	字的个数	字节的个数	第 1 个字的值	••••••••••	CRC 16
1 字节	1 字节	2 字节	2 字节	1 字节	2 字节		2 字节

应答:

从设备	10	第 1 个字的号码	字的个数	CRC 16
1 字节	1 字节	2 字节	2 字节	2 字节

实例：把值 1 和 2 写到 11 号从设备中的字 %MW16 和 %MW17 中

请求	0B	10	0010	0002	04	0001	0002	CRC16

答应	0B	10	0010	0002	CRC 16

7.4.5-7 LRC 的计算

LRC: 将帧的内容，除去头代码，用十六进制表示，求和，模 FF，然后取补码，以 ASCII 码表示。

举例:

ASCII 帧

3A	30 31	30 38	30 30 30 30 36 31 36 32	33 34	0D 0A

等同的二进制帧

01	08	00 00 61 62	4872

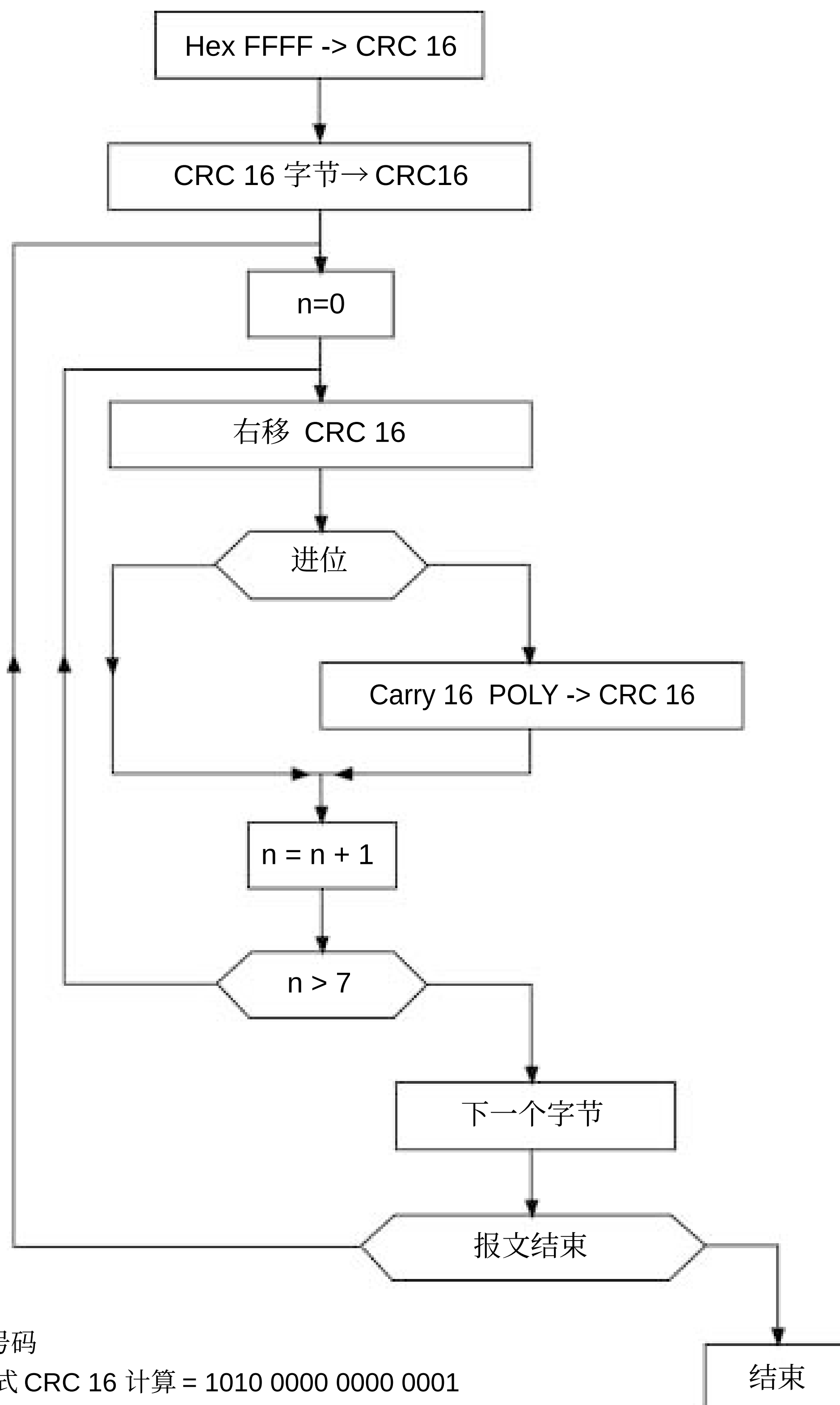
LRC 的计算

将帧内容的用十六进制求和，模 FF :

01+08+00+00+61+62=CC_H = 1100 1100

取反:	0011	0011	
加 1:		+ 1	
	0011	0100	
十六进制变换	3	4	
ASCII 码	33	34	**LRC** = 3334

7.4.5-8 CRC 16 的算法



n= 数据位的号码

POLY = 多项式 CRC 16 计算 = 1010 0000 0000 0001

(多项式发生器 = 2 = X^2 = X^{15} = X^{16})

在 CRC 16 中,发送的第一个字节是低位字节。

7.4-6 限制

TSX Neza Slave 在 MODBUS 模式下有以下限制：

- 物理层：只能是 RS485，不能根据 Master（主设备）的速度自动调节。
- Slave（从设备）间通信：从设备间不允许直接通信，只能通过主设备进行通信。
- 不允许从 ASCII 模式到 RTU 模式进行动态变化。

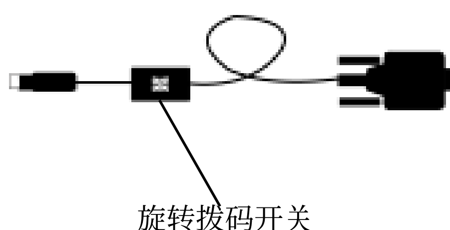
7.5 TSX08PRGCAB 多用途编程及通讯电缆

TSX08PRGCAB 是用于 RS485 与 RS232C 转换的多用途通讯电缆。

TSX08PRGCAB 可用于 Modicon TSX Neza 与多种不同的 RS232C 串行设备进行通讯。

TSX Neza 有一个带有内部供电的 RS-485 内置 TER 终端口。
如上位计算机与TSX Neza 通讯, 用于编程或监控, 则需要使用TSX08PRGCAB 电缆将RS-485 转换到 RS-232C。

同样，TSX08PRGCAB 也可以与 TSX Premium、TSX Micro 及 TSX Neza 的 TER 口相连。



型号	TSX08PRGCAB(3m)
技术指标	支持 TSX Neza TER 端口 支持 TSX Premium TER 端口 支持 TSX Micro TER 端口 支持 TSX Nano TER 端口 支持与 ACC01 接线盒连接 一个 8 pin mini DIN 针连接器 一个 9 pin D 型孔连接器 一个 4 位置旋转拨码开关
环境	工作温度 0-60°C

TSX08PRGCAB 配有一个“ 0-3 ” 4 个位置的旋转拨码开关，用于选择不同的工作模式。

TSX08PRGCAB 对于RS-485的接收／发送控制有两种选择：
由 PLC 或由 RS-232C 接口的 RTS 信号控制。
由 PLC 控制：PLC 与 TSX08PRGCAB 需直接连接；
由RS232C接口的 RTS 信号控制：PLC 与TSX08PRGCAB 可间接连接，如通过ACC01连接。

TSX08 Neza 的终端 TER 口有一个 DTP 信号引脚，用于确定该终端口的工作方式。如果 DTP 为高电平，则TER口总是工作在Uni-Telway master 方式，对PLC 编程需在该方式下。如果DTP 为低电平，则 TER 口的工作模式由 PLC 配置决定。

综合上述两项，旋转拨码开关 0~3 对应的工作模式为：

拨码位置	功能	DTP 电平	RS-485 收发控制
0	用于编程和通讯时作 Uni-Telway master（相当于 TSXPCU）	1	RTS
1	用于除 Uni-Telway master 外的其它通讯方式	0	RTS
2	用于编程和通讯时作 Uni-Telway master, PLC 与电缆需直接连接（相当于 TSXPCX M On）	1	PLC
3	用于除 Uni-Telway master 外的其它通讯方式，PLC 与电缆需直接连接（相当于 TSXPCX M Off）	0	PLC

特点：
- 容易使用、多功能、旋转拨码开关选择。
- 同时适用于 TSX Neza、Premium、Micro 和 Nano。
- 在使用不同通讯应用时只需一种电缆，不再需要多种不同的电缆。

8.附录

8.1 布尔指令列表的等价梯形图

指令列表	等价梯形图	描述
LD, LDN, LDR, LDF	P N	载入
ST, STN, R, S	( ) (/) (R) (S)	存储
AND, ANDN, ANDR, ANDF	/ P N	逻辑位 ” 与 ”
OR, ORN, ORR, ORF	/ P N	逻辑位 ” 或 ”
AND(, OR((8 层嵌套)		圆括号
XOR, XORN, XORR, XORF	XOR XORN XORR XORF	异或
N	不可逆	逻辑 “ 非 ”
END, ENDC,ENDCN ENDCN 不可逆	< END > < END >	程序结束
%Li		标号定义
JMP, JMPC, JMPCN JMPCN 不可逆	>> %Li >> %Li	跳转到标号 0 ≤ i <16
SRn	>> SRi	子程序调用
RET	< RET >	返回
NOP	不可逆	空操作

8.2 PL707WIN 编程软件变量

变量类型	描述	读／写／强置
计数器	(%C0-%C15)	
%Ci.P 预置值		
%Ci.V 当前值		
%Ci.E 下溢输出（空）		
%Ci.D	到达预置值输出	
%Ci.F溢出输出（满）		
鼓型控制器	(%DR0-%DR3)	
%DRi.S	当前步数	
%DRi.F	满（到达最后步）	
高速计数器	%FC	
%FC.P	预置值	
%FC.V	当前值	
%FC.S0	阈值 S0	
%FC.S1	阈值 S1	
%FC.TH0	阈值位 0	
%FC.TH1	阈值位 1	
%FC.F	溢出输出	
输入位	(%Ii.j 其中 i=0-3 j=0-15)	
输入字	(%IWj.0-%IWj.3 其中 j=1,2,3,4,5)	

变量类型	描述	读／写／强置
常量字	(%KW0-%KW63)	R
LIFO/FIFO 寄存器	(%R0-%R3)	
%Ri.O	输出字	R
%Ri.I	输入字	R,W
%Ri.E 寄存器空		R
%Ri.F 寄存器满		R
内部位	(%M0-%M127)	R,W
内部字	(%MW0-%MW512)	R,W
报文	(%MSG)	
%MSG.E	通信错误输出	R
%MSG.D	通信完成输出	R
脉冲发生器	(%PLS)	
%PLS.P	预置值	R,W (2)
%PLS.N	脉冲数	R,W
%PLS.Q	当前脉冲发生输出	R
%PLS.D	脉冲发生完成输出	R
脉宽调制	(%PWM)	
%PWM.P	预置周期	R,W
%PWM.R	占空比	R,W
输出位	(%Qi.j.其中 i=0-3 j=0-11)	R,W,F(3)

变量类型	描述	读／写／强置
输出字	（%QWj.0-%QWj.1 其中 j=1,2,3,4,5）	R,W
系统位	（%S0-%S127）	R,W(1)
移位寄存器	（%SBRi.0-%SBRi.15，其中 i=0-7）	R
步进计数器	（%SCi.0-%SCi.255，其中 i=0-7）	R
系统字	（%SW0-%SW127）	R,W(1)
定时器	（%TM0-%TM31）	
%TMi.V	当前值	R
%TMi.P	预置值	R,W(2)
%TMi.Q	定时到，输出	R

(1) 某些位和系统字是不能写的。在指令列表／梯形图编辑器中，没有特别的警告信息警告用户。但是写这些位或字对 PLC 有影响。

(2)如果在配置中，选择了“调整”选项，将可写这些变量。
如果没有选“调整”选项,数据编辑器的写访问将显示信息"PLC 操作不允许"。

(3)运行／停止输入或 PLC 状态（安全）输出由编程软件配置决定，且不受PLC程序的影响。

注意：对于所有不能写的变量，在数据编辑器中会对用户发警告信息。

8.3 安全特性

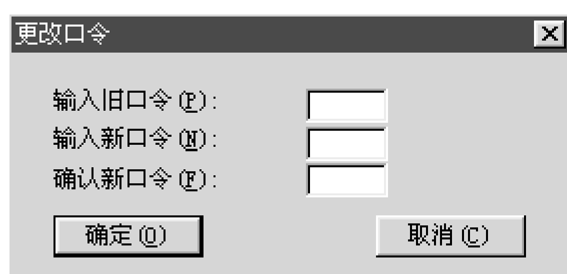
PL707WIN 编程软件的安全特性包含密码保护，用来防止不被授权的应用程序的修改和安全级的修改。在定义密码时，用户限制通过创建下面两级安全来限制对应用程序的访问：

- 管理员级 - - 可编辑应用程序的任何部分。
- 操作员级 - - 可编辑变量和数据页，但不能修改应用程序和配置数据。应用程序在打开时，默认为操作员级密码保护。

8.3-1 第一次输入密码

1.打开要设密码的应用程序

2.在文件菜单选“ 安全设置 ”，在次级菜单选“ 更改口令 ”。出现改变密码对话框。



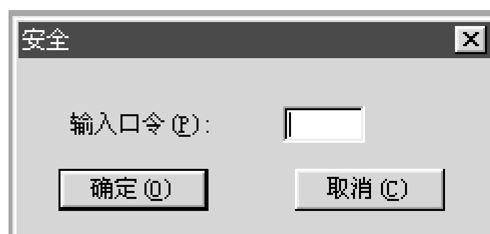
3. 在“ 输入旧口令 ” 域，不要输入任何值，在“ 输入新口令 ” 域，输入长度为 1 到 8 字符间的密码。
4. 在“ 确认新口令 ” 域，重新输一遍密码。
5. 选“ 确定 ”，使用新密码，应用程序标题栏显示“ 管理员级 ”。
   选“ 取消 ” 关闭对话框，没有定义密码。
6. 应用程序必须存盘，新密码才起作用。

8.3-2 改变密码

1. 在文件菜单选“ 安全设置 ”，在次级菜单选“ 更改口令 ”，出现对话框。
2. 在“ 输入旧口令 ” 域，输入旧密码，在“ 输入新口令 ” 域，输入长度为 1 到 8 字符的密码。
3. 在“ 确认新口令 ” 域，重新输入一遍密码。
4. 选“ 确定 ”，使用新密码。选“ 取消 ” 关闭对话框，不改变密码。
5. 应用程序必须存盘，新密码才起作用。

8.3-3 从操作员级改为管理员级

1. 在文件菜单选“ 安全设置 ”，在次级菜单中选“ 管理员级 ” ，出现安全对话框。



2. 输入密码。选“ 确定 ” 改到管理员级。选“ 取消 ” 回到操作员级。

8.3-4 删除密码

1. 要删除密码，在文件菜单中选“ 安全 ”，在次级菜单中选“ 更改口令 ”。
2. 在改变密码对话框，在“ 输入旧口令 ” 对话框，键入当前密码。
3. 到“ 输入新口令 ” 和“ 确认新口令 ” 域，把这些域的值置为空。
4. 选“ 确定 ”，删除了密码。选“ 取消 ” 回到应用程序，不改变密码。

8.4 指令扫描时间和内存占用量

时间以微秒表示。
大小以字节表示。

布尔指令

指令	以位计算			
	%I,%Q		%M, %S, 0/1, %MWi:Xi... (1)	
	时间	所占空间	时间	所占空间
LD, LDN	0.2	2	0.4	4
LDR, LDF	0.5	4	-	-
AND,ANDN	0.2	2	0.6	4
ANDR,ANDF	0.8	4	-	-
AND(,AND(N	6.3	8	7	10
AND(R,AND(F	7	10	-	-
OR,ORN	0.4	2	0.7	4
ORR, ORF	0.8	4	-	-
OR(,OR(N	6.3	8	6.6	10
OR(R, OR(F	6.7	10	-	-
XOR,XORN	0.2	2	0.6	4
XORR,XORF	0.8	4	-	-
ST,STN	0.2(0.8)	2	0.9(1.5)	4
S,R	0.8(1.2)	4	1.5(1.8)	6
N	0.6	4		
)	15	8		
MPS	0.6	4		
MRD	0.2	2		
MPP	0.2	2		

写在程序行 099 后的指令，时间要乘以 3。

(1) 对字抽取位 %MW16 到 %MW255 的字和其它类型(%KWi:Xj,%SWi:Xj)，这些时间要乘以 1.5，大小要加 2 字节。

功能块指令（在可逆化编程中）

指令	扫描时间（微秒）	内存占用量（字节）
BLK %TMi	8	4
BLK %Ci	8	4
BLK %Ri	8	4
BLK %SBRi	8	4
BLK %SCi	8	4
BLK %DRi		
BLK %FC	8	4
BLK %MSG	8	4
BLK %PMW	8	4
BLK %PLS	8	4
OUT_BLK	200	2
END_BLK	180	2
IN	1.2	4
R	0.6	4
CU	0.7	4
CD	0.7	4
I	1	4
O	1	4
U		
S	0.7	4

功能块指令（在不可逆编程中）		
指令	扫描时间（微秒）	内存占用量（字节）
IN %TMi	48	4
CD %Ci (CU %Ci)	46	4
S %Ci	49	4
R %Ci	47	4
U %DRi		
R %DRi		
LD %SCi.j	9	6
CD %SCi (CU %SCi)	38	4
ST %SCi.j	10	6
R %SCi	36	4
BLK %PMW	42	4
BLK %PLS	53	4
CD %SBRi (CU %SBRi)	39	4
R %SBRi	37	4
I %Ri (O %Ri)	49	4
R %Ri	48	4
IN %PWM	36	4
IN %PLS	46	4
S %PLS	42	4
R %PLS	58	4
IN %FC	43	4
S %FC	69	4
READ	9.8	6
EXCH	160-700	8
R %MSG	25	4

数字指令

指令	扫描时间（微秒）	内存占用量（字节）
:=	29.5	10
+	34	12
-	38	12
*	49	12
/	48	12
REM	49	12
INC	28	6
DEC	28	6
AND	37	12
OR	37	12
XOR	37	12
NOT	29	8
SHL	34	10
SHR	34	10
ROL	35	10
ROR	35	10
BTI	40	8
ITB	40	8
SQRT	80	8

比较指令

指令	扫描时间（微秒）	内存占用量（字节）
LD[word1 comp word2]	18	8
AND[word1 comp word2]	19	10
AND([word1 comp word2]	24	14
OR[word1 comp word2]	21	10
OR([word1 comp word2]	25	14

comp : 比较操作 => ， <= ， <> ， = ， >= ， >

程序指令

指令	扫描时间（微秒）	内存占用量（字节）
END	0.4	2
ENDC, ENDCN	0.6	4
SR	14	4
RET	2	6
NOP	0.4	2
JMP	7.8	4
JMPC, JMPCN	8	6
%Ln :	0.6	4
%SRn :	2	4

8.5 导入导出 ASCII 程序文件和变量文件

PL707WIN 编程软件能够导入导出 ASCII 程序文件和变量文件。

8.5-1 ASCII 程序文件

列表程序可由外部 ASCII 文本编辑器开发。ASCII 格式程序导入导出功能允许文件能在 PL7-07 软件和 ASCII 文本编辑器之间传送文件。

导入 **ASCII** 程序文件：

1.打开一个新的或已经存在的应用程序，从文件菜单选“ 导入 ”。从导入次级菜单选“ ASCII程序 ”。出现文件选择对话框。



2.在“ 搜寻 ” 域，选文本文件所在的驱动器和路径。
3.在“ 文件类型 ” 域，选文本（*.txt）或所有（*.*）来显示路径下的文件。
4. 在“ 文件名 ” 域，选要导入的文本文件。
5.选“ 确定 ”，导入文件。选“ 取消 ”，关闭对话框，不导入文件。

导出 **ASCII** 程序文件：

1. 打开一个应用程序，从文件菜单选“ 导出 ”，从次级菜单中选 “ ASCII 程序 ”，出现文件选择对话框。
2.在“ 保存类型 ”域，选文本（*.txt）或所有（*.*）。.txt 文件扩展名指明是文本编辑器文件。
3. 在“ 保存在 ” 域，选要存入文件的驱动器和路径。
4.在“ 文件名 ” 域，键入一个标准 WINDOWS 文件名。
   如果文件名与 WINDOWS 文件命名准则冲突，会出现 “ 无效文件名 ” 信息。
   如果输入的文件名与该路径下已经存在的文件的文件名重复，出现错误信息："选择的文件已经存在。覆盖？” 。选 ”确定“ 或 ”取消 ”。
5. 选“ 确定 ”，导出文件。选“ 取消 ”，关闭对话框，不导出文件。

8.5-2 变量文件

可以为普通应用程序建立变量文件。变量导入导出让用户可以在应用程序之间传送变量。

导入变量文件:

1.打开一个新的或已经存在的应用程序，从文件菜单中选 “ 导入 ”，从次级菜单中选 “ 变量 ”，出现文件选择对话框。



2. 在 “ 搜寻 ” 域，选变量所在的驱动器和路径。
3. 在 “ 文件类型 ” 域，选变量（*.sym）或所有（*.*）来显示路径下的文件。
4. 在 “ 文件名 ” 域，选要导入的变量文件名。
5. 选 “ 确定 ”，导入文件。选 “ 取消 ”，关闭对话框，不导入文件。
6. 如果在第 5 步中选的是 “ 确定 ”，状态栏中会出现信息。

导出变量文件:

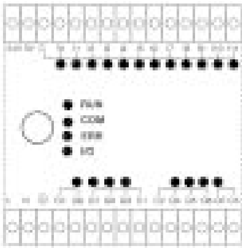
1. 打开一个应用程序，从文件菜单中选 “ 导出 ”。在导出次级菜单中选 “ 变量 ”，出现文件选择对话框。
2. 在 “ 保存类型 ” 域，选变量（*.sym）或所有（*.*）。.sym 文件扩展名表明是变量文件。
3. 在 “ 保存在 ” 域，选要存入文件的驱动器和路径。
4. 在 “ 文件名 ” 域，键入标准 WINDOWS 文件名。
   如果文件名与 WINDOWS 文件命名准则不符，会出现 “ 无效文件名 ” 消息。
   如果文件名与已存在于该路径下的文件名相同, 会出现错误信息:“ 选择的文件名已存在。覆盖？ ”，选 “ 确定 ” 或 “ 取消 ”。
5. 选 “ 确定 ”，导出文件。选 “ 取消 ”，关闭对话框，不导出文件。
6. 如果在第 5 步中选的是 “ 确定 ”，在状态栏会出现信息。

8.6 故障定位与分析

8.6-1 使用 PLC 面板上的 LED 状态分析故障

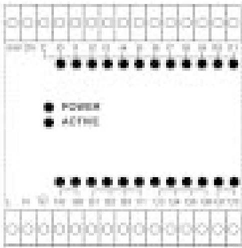
• 显示 **PLC** 的状态

本体 Neza PLC 模块将连续执行自检的结果通过面板上的 4 个指示灯显示，这 4 个指示灯为：RUN、ERR、I/O 和 COM。



LED	灭	闪烁	亮
RUN（绿）	没有电源或硬件故障	PLC STOP	PLC RUN
COM（黄）	没有通讯	Modbus，Uni-Telway，ASCII 通讯	远程 I/O
ERR（红）	运行 OK	用户应用程序错误	硬件故障
I/O（红）	运行 OK	Block I/O 扩展 OK	扩展 I/O 模块有故障

扩展 Neza PLC 模块将执行状态通过面板上的 2 个指示灯显示，这 2 个指示灯为：POWER、ACTIVE。



LED	灭	亮
POWER（绿）	没有连接或硬件故障	已与本体连接
ACTIVE（绿）	本体没有运行或硬件故障	本体运行 I/O 扩展 OK

8.6-2 使用系统位和系统字分析故障

如果检测到一个故障, PLC 将对相应的系统位或系统字置位。可以在应用程序中使用这一信息。TSX08PALM 编程终端或 PL707WIN 数据编辑器可以显示系统位（%S）和系统字(%SW)。

8.6.2-1 系统位

系统位	功能	描述
%S10	I/O 故障	正常值为 1。当检测到主 PLC 或同级 PLC 上的 I/O 故障（配置错误、交换错误、硬件故障、受保护的固态输出断开）时，将其置为 0。位 %S118 和 %S119 显示故障在哪一个 PLC 上，字 %SW118 和 %SW119 描述了错误的详细内容。当故障排除时，%S10 复位到状态 1。
%S11	警戒时钟超时	正常值为 0，当程序的执行时间（扫描时间）超过最大扫描时间（150 ms 软件警戒时钟）时，系统将其置为 1。警戒时钟超时将导致 PLC 停止。
%S19	扫描周期超时（定期任务）	正常值为 0, 当扫描周期超时(扫描时间大于用户在配置中定义的或在 SW0 中设定的时间)，系统将其置为 1。此位必须由客户复位到 0。
%S71	通过扩展连接交换	初始值为 0。当检测到通过扩展连接交换时置为 1, 当扩展连接没有交换时置为 0。主 PLC 的字 %SW71 显示有效扩展的清单和状态。
%S118	主 PLC 上的 I/O 故障	正常值为 0，当检测到主 PLC 上的 I/O 故障时置 1。%SW118 字给出了错误的详细内容。当故障排除时，%S118 复位为 0。
%S119	对等 PLC 上的 I/O 故障	正常值为 0，当检测到对等 PLC 上的 I/O 故障时置 1。%SW119 字给出了错误的详细内容。当故障排除时，%S119 复位为 0。

8.6.2-2 系统字

系统字	功能	描述
%SW71	PLC 通讯连接上的设备	显示每一个同级 PLC 和 I/O 扩展与主 PLC 通讯的状态: 第 1 位：I/O 扩展 第 2 位：第 2 个同级 PLC 第 3 位：第 3 个同级 PLC 第 4 位：第 4 个同级 PLC 如果没有同级 PLC 或 I/O 扩展、没有接通电源、没有进行电缆连接或出现故障时，对应位为 0。 当有同级 PLC 与主 PLC 通讯时，对应位为 1。
%SW118	主 PLC 状态	显示检测到的主 PLC 上的故障: 第 1 位为 0：其中一个输出端故障； 第 3 位为 0：传感器电源故障； 第 8 位为 0：TSX 07 内部故障或硬件故障； 第 9 位为 0：外部或通讯故障； 第 11 位为 0：PLC 执行自检； 第 13 位为 0：配置错误； 这个字的其它所有位都为 1，且保留未用。因此，对于一个没有任何故障的 PLC，这个字的值为：16#FFFF。
%SW119	同级 PLC 状态	显示检测到的同级 PLC 上的故障（这个字仅由主 PLC 使用）。这个字各个位的分配和字 %SW118 几乎完全一样，除了： 第 13 位：无意义， 第 14 位：初始化时存在的同级 PLC 现在丢失。

B

施耐德电气公司
Schneider Electric China
www.schneider-electric.com.cn

北京市朝阳区东三环北路 8 号
亮马大厦 17 层
邮编: 100004
电话: (010) 6590 6907
传真: (010) 6590 0013

17/F, Landmark Building
8 North Dongsanhuan Road
Chaoyang District Beijing 100004
Tel: (010) 6590 6907
Fax: (010) 6590 0013

由于标准和材料的变更, 文中所述特性和本资料中的图象只有经过我们的业务部门确认以后，才对我们有约束。



本手册采用生态纸印刷

SC DOC 563-Neza

2002.10